A STATE OF THE ART REPORT

ELECTRONIC PUBLISHING
ON THE
WORLD WIDE WEB
AN ENGINEERING APPROACH

DACS Contract #F30602-92-C-0158
TECHNICAL AREA TASK 30

Prepared for:

Defense Technical Information Center
8725 John J. Kingman Road Ste 0944
Fort Belvoir, VA 22060-6218

Prepared by:

Elaine Fedchak, Lorraine Duvall,
James DeLude, Alan Piszcz, Robert Vienneau

Kaman Sciences Corporation
258 Genesee Street
Utica, NY 13502

29 September 1995

19990604 039

Data & Analysis Center for Software
P.O. Box 120
Utica, NY 13503-0120

DACS

DTIC QUALITY INSPECTED 4

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 29 September 1995	3. REPORT TYPE AND DATES COVERED 28 Sept. 1994 - 29 Sept. 1995

4. TITLE AND SUBTITLE State of the Art Report Electronic Publishing on the World Wide Web, An Engineering Approach	5. FUNDING NUMBERS F30602-92-C-0153

6. AUTHOR(S)
Elaine Fedchak, Lorraine Duvall, James DeLude, Alan Piszcz, Robert Vienneau

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Kaman Sciences Corporation 258 Genesee Street Utica, NY 13502	8. PERFORMING ORGANIZATION REPORT NUMBER N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Sponsoring Org. Monitoring Org. Defense Technical Info Center Rome Laboratory DTIC/AI 8725 John J. Kingman Rd. RL/C3C Ft. Belvoir, VA 22060-6218 Griffiss AFB, NY 13441	10. SPONSORING/MONITORING AGENCY REPORT NUMBER N/A

11. SUPPLEMENTARY NOTES

Available from: Data & Analysis Center for Software
P.O. Box 120, Utica, NY 13503

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release. Distribution unlimited	12b. DISTRIBUTION CODE UL

13. ABSTRACT (Maximum 200 words)

Electronic publishing refers to the use of computer technology in publishing or distributing information. This report focuses on electronic publishing using the World Wide Web (WWW) as a distribution medium. The WWW is an Internet service that combines hypertext capabilities with information discovery techniques, allowing users to access hypermedia information remotely. The theme of this report is that electronic publishing on the Internet is in many ways analogous to developing software. Therefore, the principles and practices of software engineering that have emerged from many years of learning how best to create and maintain software can be usefully applied to electronic publishing. Electronic documents developed from an engineering perspective will meet the growing demand for quality information products at a manageable cost.

14. SUBJECT TERMS World Wide Web, Electronic Publishing, Hypertext, Software Engineering, Hypertext Markup Language, Authoring, Kiosks	15. NUMBER OF PAGES 165
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT unclassified	20. LIMITATION OF ABSTRACT UL

PREFACE

The purpose of this handbook is to provide guidelines to World Wide Web (WWW) information providers for disseminating information through the WWW. This version of the handbook is a Data & Analysis Center for Software (DACS) State of the Art Report (SOAR). A hypertext version of the handbook is available on the World Wide Web at the DACS home page (http://www.utica.kaman.com). The initial draft of this handbook was developed as part of a DACS Technical Area Task, in support of an effort to develop Internet-accessible products.

Product names mentioned in this handbook may be trademarks; they are used for identification purposes only.

The DACS is a Department of Defense (DoD) Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC) under the DoD IAC Program. The DACS is monitored by Rome Laboratory (RL) and operated by Kaman Sciences Corporation. The DACS serves as a centralized source of current, readily available data and information concerning software engineering and software technology.

The DACS can be contacted at:

> Data & Analysis Center for Software
> Kaman Sciences Corporation
> PO Box 120
> Utica, NY 13503-0120
> Phone: (315) 734-3696
> FAX: (315) 734-3699
> E-mail: dacs@utica.kaman.com
> Home page: http://www.utica.kaman.com

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

This handbook is a guide to disseminating information through the Internet. Its primary focus is the construction and publication of documents on the World Wide Web (WWW). The handbook incorporates lessons learned by the Data & Analysis Center for Software (DACS) from experience gained in using emerging Internet capabilities, particularly the World Wide Web.

Electronic publishing in general refers to any use of computer technology in publishing or distributing information. This handbook, however, focuses on electronic publishing using the WWW as a distribution medium. The WWW, also known as W3 or the Web, is an Internet resource discovery service that combines hypertext capabilities with information discovery techniques, allowing users to access hypertext information remotely.

The underlying theme of this handbook is that electronic publishing on the Internet is in many ways analogous to developing software. The principles and practices of software engineering that have emerged from many years of learning how best to create and maintain software can be usefully applied to electronic publishing. Electronic documents developed from an engineering perspective will meet the growing demand for quality information products at a manageable cost.

1.1 Scope

The scope of this handbook is to provide guidelines for authoring and maintaining information to be disseminated via the World Wide Web. High level design issues; implementation styles and options; maintenance implications; and management issues are included. The intended audience is Web information providers. The purpose of the handbook is to define a process that can be used for electronic publishing activities, rather than to duplicate existing sources of information. More detailed technical explanations can be found in the referenced materials.

The handbook was initially developed as part of a DACS Technical Area Task in support of an effort to investigate the use of the WWW by Information Analysis Centers (such as the DACS) for distributing information to their users. That task included the development of Internet-accessible products. The handbook has since evolved into more general guidelines for Internet information providers, incorporating results and experiences from that Technical Area Task, including the conversion of the handbook itself for publication on the Web.

1.2 How to Use the Handbook

This handbook has been designed to guide multiple audiences in pursuing an engineering approach to WWW publishing. It supports several levels of experience and several objectives. Roadmaps for five potential combinations of users and uses have been developed using the *Table of Contents* as a template. The identified users and uses include:

1. Quick Start: Novices who want a quick overview, just to get started

2. Improve Quality: Experienced Web users who want to improve the quality of their electronic documents

3. Kiosk Development: Organizations who want to develop a presence on the WWW

4. Interest in Multimedia: Information providers who want to add multimedia to their electronic documents

5. Explore Alternatives: Newcomers who want to explore the many possibilities and potentials of electronic publishing

An analogy between electronic publishing and software engineering is introduced in the discussion of a document life cycle. The analogy is recalled throughout the handbook to illustrate parallels between electronic publishing and software development. Readers

who have some familiarity with software development or management will find the analogy useful for increasing their understanding of electronic publishing issues. It is not, however, necessary to be a software expert to use or understand the handbook.

1.2.1 Quick Start

The Quick Start perspective is for people new to electronic publishing, who want to put up only a page or two. The sections of the handbook highlighted in Figure 1 are the minimum needed to get started. The background information defines basic terminology used in WWW publishing, which is necessary to understand the other how-to sections.
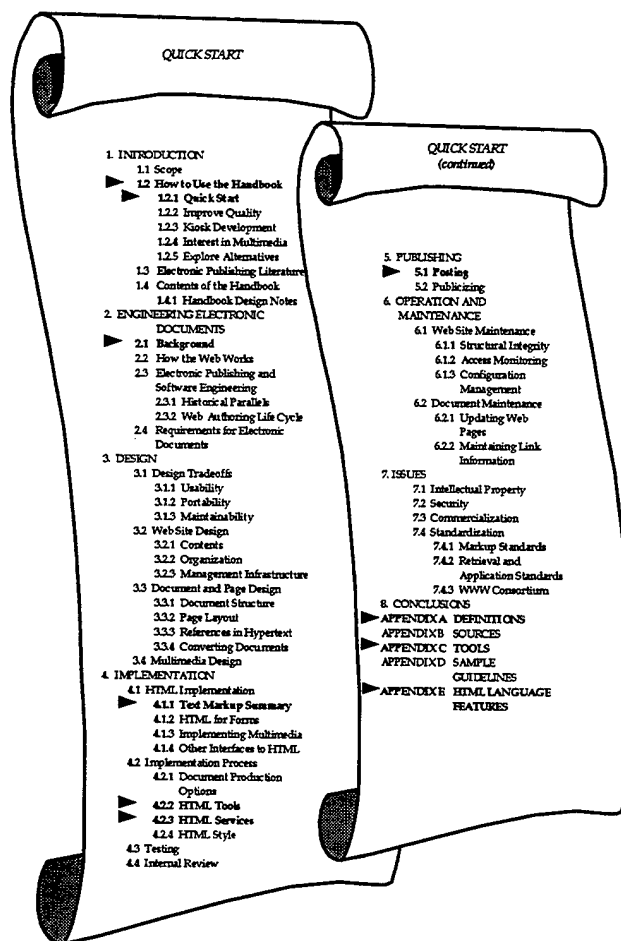
1.2.2 Improve Quality

The Improve Quality perspective is for people who are familiar with the basics of electronic publishing but want to make their Web documents better, and the creation and maintenance processes more productive. The handbook sections highlighted in Figure 2 skip the introductory and background material, pointing instead to the design and maintenance considerations that are not usually found in Web publishing documentation. This perspective emphasizes electronic publishing as an engineering activity, and therefore as one that benefits from organization and planning, articulated objectives, a well-defined process, and discipline.

Figure 1: Quick Start
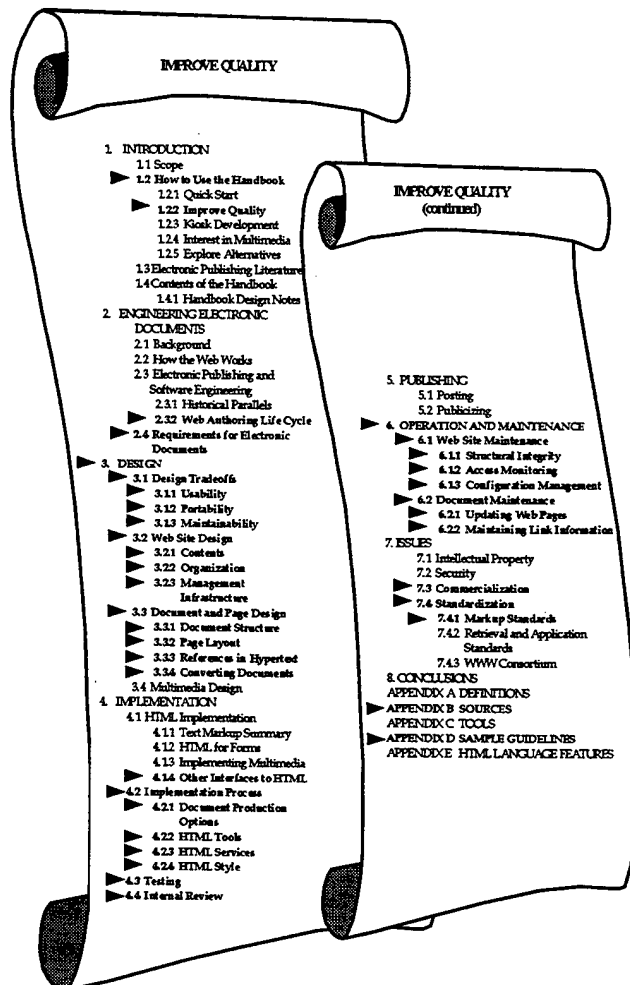
Figure 2 : Improve Quality

Data & Analysis Center for Software

1.2.3 Kiosk Development

The Kiosk Development perspective is for those who want to build a comprehensive Web site, to organize and publicize information resources related to a specific topic or group. The term kiosk is becoming a common name for describing a Web site that provides comprehensive, coherent, and useful information. Like its real-world equivalent, an Internet kiosk is an information booth designed for public access. It contains introductory and background information, answers to common questions, and pointers to additional information. The goals of developing a kiosk are to establish an organizational presence on the Web and to facilitate communication, either internally among the people of an organization, or externally with customers and the general public.
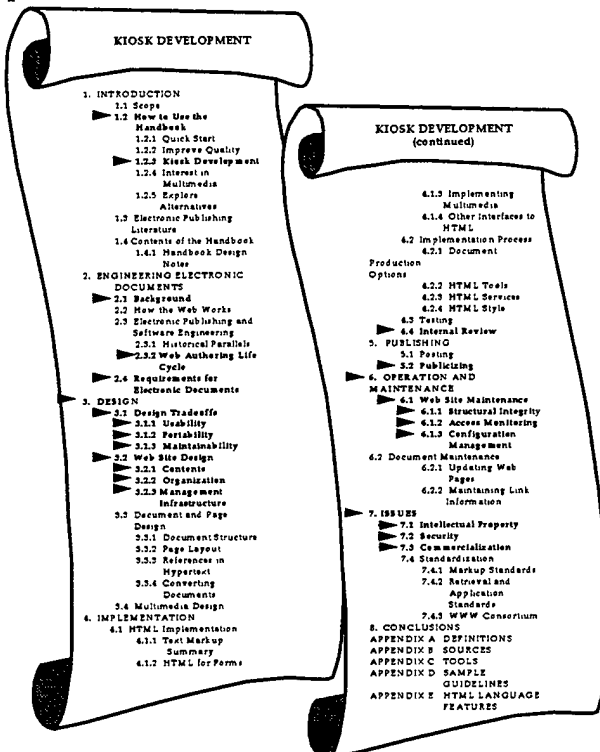
Figure 3: Kiosk Development

Successful kiosk development requires understanding of the higher-level design and management issues surrounding electronic publishing, in addition to the technical and implementation details. Handbook sections of interest for kiosk development are highlighted in Figure 3. After the content and structure of the kiosk have been defined, and planning for its implementation has been completed, the guidelines in the rest of the handbook will be applicable to those tasked to carry out the plan.

1.2.4 Interest in Multimedia

The Interest in Multimedia perspective is for people who are adding multimedia components to their electronic documents, or improving the quality of the non-textual information available on their Web sites. The two sections *Multimedia Design* (Section 3.4) and *Implementing Multimedia* (Section 4.1.3) are the primary handbook sections that provide guidance for multimedia. Other topics related to multimedia on the Web are included within the sections *Design Tradeoffs* (Section 3.1), *References in Hypertext* (Section 3.3.3), and *Intellectual Property* (Section 7.1).

1.2.5 Explore Alternatives

The Explore Alternatives perspective is for people with minimal Web experience, who are interested in learning about different capabilities and experimenting with various authoring and publishing techniques. This is the primary audience and purpose for which the handbook was developed; therefore, the entire contents can be considered applicable.

1.3 Electronic Publishing Literature

As prelude to developing this handbook, a literature search of both the Internet and traditional reference materials was undertaken. Acquisition and review of new sources continued throughout the handbook development and revision cycles. Useful and relevant source information is listed in Appendix B, sorted by topic area. The

following observations summarize results of the literature searches:

- Hard-copy references lag behind the on-line information, but are less prone to unexpected disappearance
- Documents on the Web range widely in quality, readability, usefulness
- There is consensus about the proper way to implement some constructs in Web documents, as well as much debate about others
- The few guidelines for designing Web pages do not necessarily agree with one another

The literature search also revealed that:

- Design considerations for portability and maintainability are seldom discussed
- There is a great need for design guidance

Thus the focus of this handbook: to gather and organize the advice and information available in multiple, disparate sources, and to fill in the gaps.

1.4 Contents of the Handbook

The guidelines in the handbook are organized as follows. Chapter 1 consists of this introduction. Chapter 2 begins with background information and terminology for understanding the WWW, then defines a life cycle approach to electronic publishing, analogous to a software development effort. Chapter 3 provides high level design issues and guidelines to increase the quality of Web documents. Chapter 4 addresses implementation issues, including an overview of the of the Hypertext Markup Language (HTML), authoring tools, reviewing, and testing electronic works. Chapter 5 describes procedures and places for announcing the availability of Web sites or products. Chapter 6 discusses maintenance and operational issues. Chapter 7 discusses issues related to electronic publishing that need to be addressed as the Internet grows and changes, including security, intellectual property, commerce, organizational image, and standards. Chapter 8 provides concluding remarks, and looks to the future of WWW publishing.

The Appendices include a list of acronyms, a brief glossary, references, a list of sources sorted by topic, a report describing the experimental use of several HTML conversion tools, sample guidelines from other organizations, and a "test page" containing the complete HTML syntax that can be used for look-ups, or to exercise a browser.

1.4.1 Handbook Design Notes

In the course of developing and describing document design issues and techniques for WWW publishing, some of the ideas leaked into the design of the handbook itself (even the paper version):

- Handling of Uniform Resource Locators (URLs):
  Throughout the handbook, explicit use of URL references has been minimized. This is done for reasons of maintainability, which are explained in more detail in the relevant sections of the handbook. The URLs of referenced Internet sources are given in Appendix B, with the last date that they were accessed. The only other use of URLs within the handbook is in tables, because they can be found easily when updating is necessary.
- Choice of examples:
  The examples used in the handbook make heavy use of pages from the DACS Web site. This is not to imply that they are the best examples, or even that they follow all the guidelines in the handbook. The reasons for using them are:
  - They are easily (i.e., locally) accessible
  - Their contents are known to the handbook authors (one of whom *is* the DACS Webmaster)
  - Use of pages and images from the DACS removes the burden of obtaining permissions to include them, and avoids any potential for inadvertent copyright infringement
- Coverage:
  The handbook focuses on the creation and update of hypertext documents. Emerging Web capabilities, such as interfaces to databases and interactive applications, are introduced but not covered in detail.

2 ENGINEERING ELECTRONIC DOCUMENTS

2.1 Background

Concepts and terms necessary to understand the guidelines in the handbook are introduced in this section. A brief glossary of the specialized terms used in the handbook is contained in Appendix A. More detailed introductions to Internet tools and capabilities can be found in the Internet information sources listed in Appendix B.

Hypertext refers to a non-linear organization of objects, such as documents, that incorporates internal and external links between related pieces of information. A comparison between traditional sequential and hypertext document organizations is depicted in Figure 4. WWW hypertext documents may contain links to other resources on the Internet. WWW links can be set up to display another document, to retrieve files, to connect interactively to a remote computer, and to access other Internet tools.

Hypertext documents on the World Wide Web are written in the Hypertext Markup Language (HTML). The term markup derives from the way proof-readers have traditionally penciled in marks that indicate how a document is to be revised. Documents on the Web are made up of separately retrievable pages, corresponding to individual files, and may include text, images, sound clips, or video. This handbook provides guidelines for constructing hypertext documents by selecting and linking files together, and adding HTML markup to files.
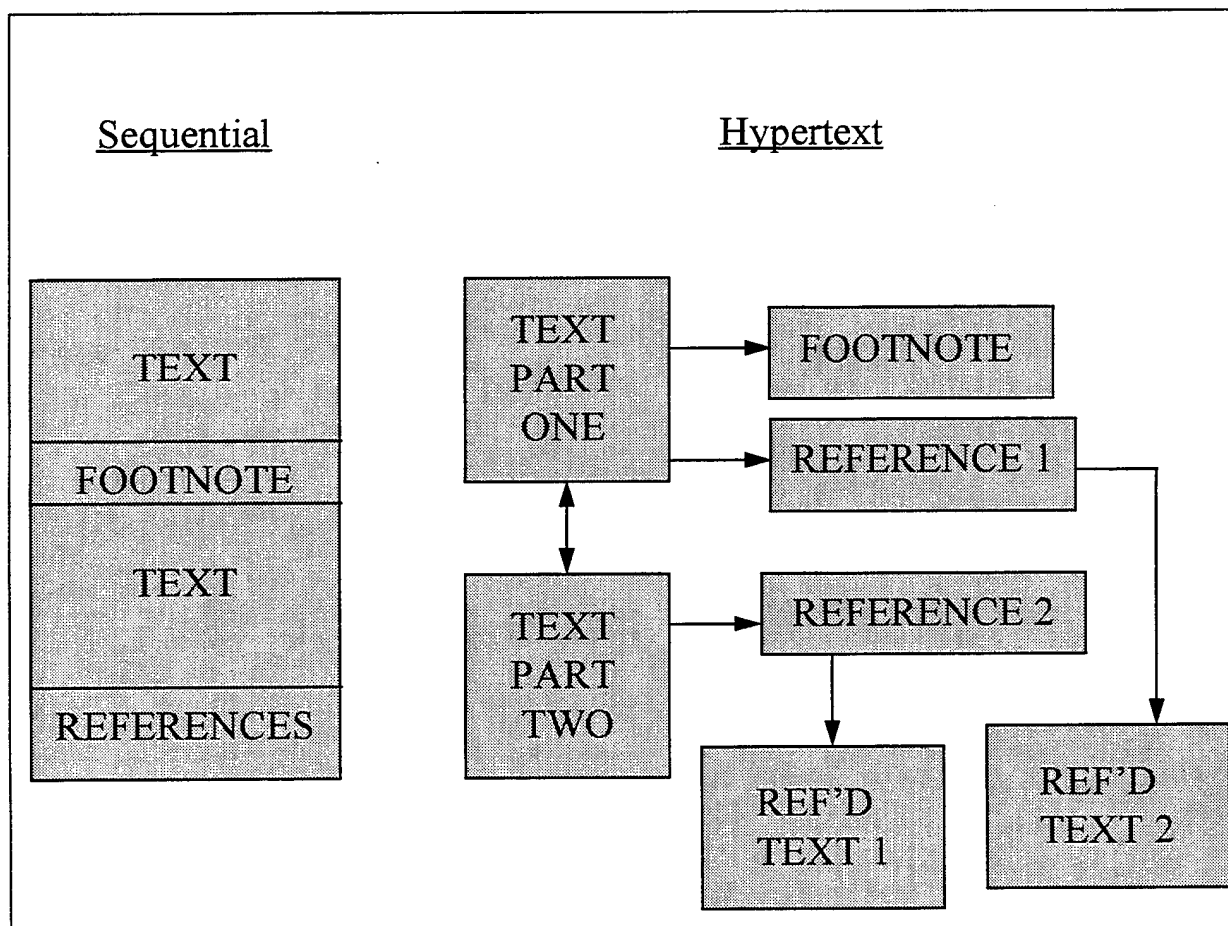
Figure 4: Contrasting Document Structures

The architecture of the WWW is client-server based. Information is published on the Web by putting HTML files on a host machine that is connected to the Internet. Each server provides access to its own documents. Documents on any one server can include links to documents on any other servers. Users access and retrieve information from the Web by using WWW client programs called browsers. Every WWW browser is required to understand HTML.

When a user selects a hypertext link[1], the browser program traverses the Internet, retrieves the referenced document or image, and displays it on the user's screen. The user, on the client side, does not need to know the location of the server's machine; the browser program determines it from interpreting the HTML coding.

The Mosaic WWW browser, from the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC), provides a Graphical User Interface (GUI) to the Web. Several other Web browsers provide a graphical interface, including Cello, Chimera, and commercial browsers, such as Netscape and Sun Microsystems' HotJava. Non-graphical browsers available to Web users include the WWW Line Mode Browser, and Lynx, which runs on VT100-compatible terminals. The previous draft of this handbook focused on use of Mosaic because at the time it was available for the widest array of platforms, and was the most commonly used browser. Since then, several commercial versions and updates of the GUI-Mosaic browser have been released. Netscape, for example, is now considered by many to be the de facto standard browser. Many of the Mosaic-specific references that appeared in the draft version of this handbook have therefore been replaced by Netscape references.

---

[1] Many browsers display the hypertext links as underlined words. Images used as links are displayed with a border around them. On some platforms, the links are highlighted with color, for example, by using blue letters or a blue border around an image.

2.2 How the Web Works

The WWW began as a project to allow remote access to hypertext information on the Internet. The WWW project research resulted in the definition of three components which interact to make the WWW possible:
- The hypertext transfer protocol (HTTP)
- The hypertext markup language (HTML), defined in Section 2.1
- The uniform resource locator (URL) address system

The Internet is a world-wide network of computer networks that use a single communications protocol, the Transmission Control Protocol/Internet Protocol (TCP/IP). The Web project developed the hypertext transfer protocol to support hypertext spanning the Internet. The WWW also supports a wide variety of existing communication protocols, and has been designed to allow for expansion to accommodate new protocols as they are invented. The hypertext transfer protocol is the transfer mechanism between client and server for WWW exchanges. Other Internet resource discovery tools which are accessible from the Web use their own protocols. These tools include the File Transfer Protocol (FTP) and Archie; Gopher and Veronica; the Wide Area Information Server (WAIS); and news services. Check the sources listed under Internet tools in Appendix B for more about these information retrieval tools.

Each computer that accesses the Internet networks has a unique numeric address, into which is coded the name of the network(s) to which it is connected, and the name of the machine itself. An example Internet address is 192.73.45.113. Most machines also have a hostname, and one or more associated nicknames, that are easier to remember than a string of up to twelve digits. The machine at the address given, for example, is also known as "www.utica.kaman.com."

Hostnames are structured to indicate, from right to left, the domain, sub-domains, and

Data & Analysis Center for Software

Hostnames are structured to indicate, from right to left, the domain, sub-domains, and individual machine name. Internet domains are both geographic and organizational. By convention, hostnames for computers in the United States end with an organizational domain name, e.g., ".com, " whereas hostnames for foreign computers end with their geographical domain, e.g., ".uk" identifies a server in Great Britain. Table 1 lists the organizational domains in use.

The Uniform Resource Locator addressing system allows for a variety of data types and protocols to be accessible on the WWW. Figure 5 provides an overview of the format for a URL. The first element specifies which protocol, e.g., *http:*, to use. Some of the possible protocols are http, ftp, gopher, nntp (the Network News Transfer Protocol), and wais. Other elements of the URL identify the host computer and the file to retrieve for downloading or display. In HTML documents, hyperlinks are created by specifying the URL for the target resource at the point in the source document where the link is to be made. Web browsers also allow a user to specify the URL of a desired resource directly, without having a predefined link to it in the current document.

Table 1: Internet Organizational Domains

Domain	Organization Type
.edu	educational
.com	commercial
.gov	government
.mil	military
.org	nonprofit
.net	network support center

http:	**//**	**www.utica.kaman.com**	**:80**	**/info.html**
access method: http, ftp, wais, nntp, gopher, etc.	indicates a machine name follows	host machine name including domain	port (usually optional)	file, including pathname

Figure 5: Components of a Uniform Resource Locator

INTERNET

HTTP
FTP
NEWS
WAIS
GOPHER

HTTP
FTP
NEWS
WAIS
GOPHER

WWW
INFORMATION
SERVER

WWW
BROWSER
(CLIENT)

HTML

CGI

HTML
Documents
:
Images

C/C++
PERL
TCL
The Bourne Shell
The C Shell
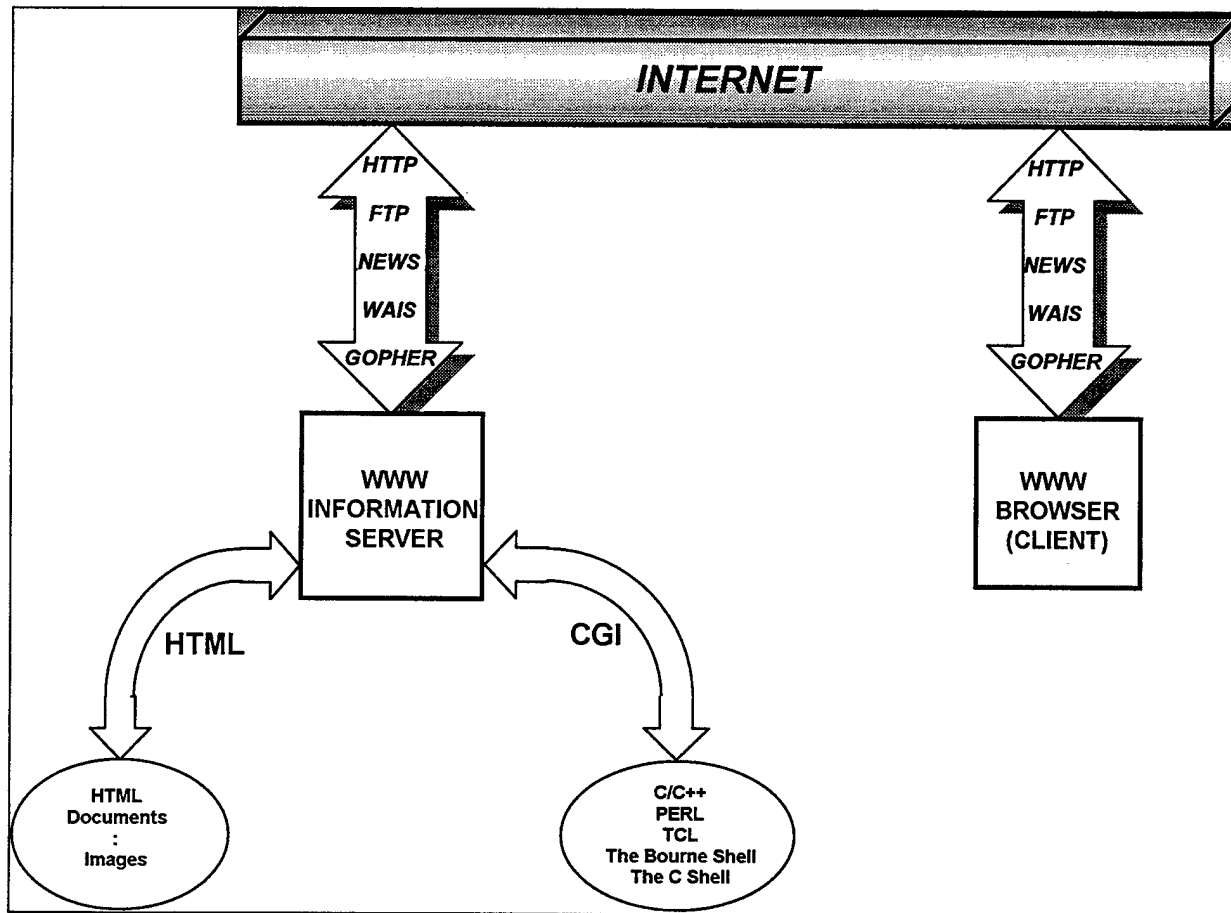
Figure 6: Connections to the Internet

Figure 6 illustrates graphically the various protocols by which client and server can connect to the Internet. Web information providers, located on the server side of the transaction, can make information available to the server in two different forms, as Figure 6 also illustrates. The first, HTML, is used for text and for creating links to multimedia objects. The other, the common gateway interface (CGI), is used to implement forms and other interactive services across the Internet. These are discussed in more detail in Section 4.1.

Data & Analysis Center for Software
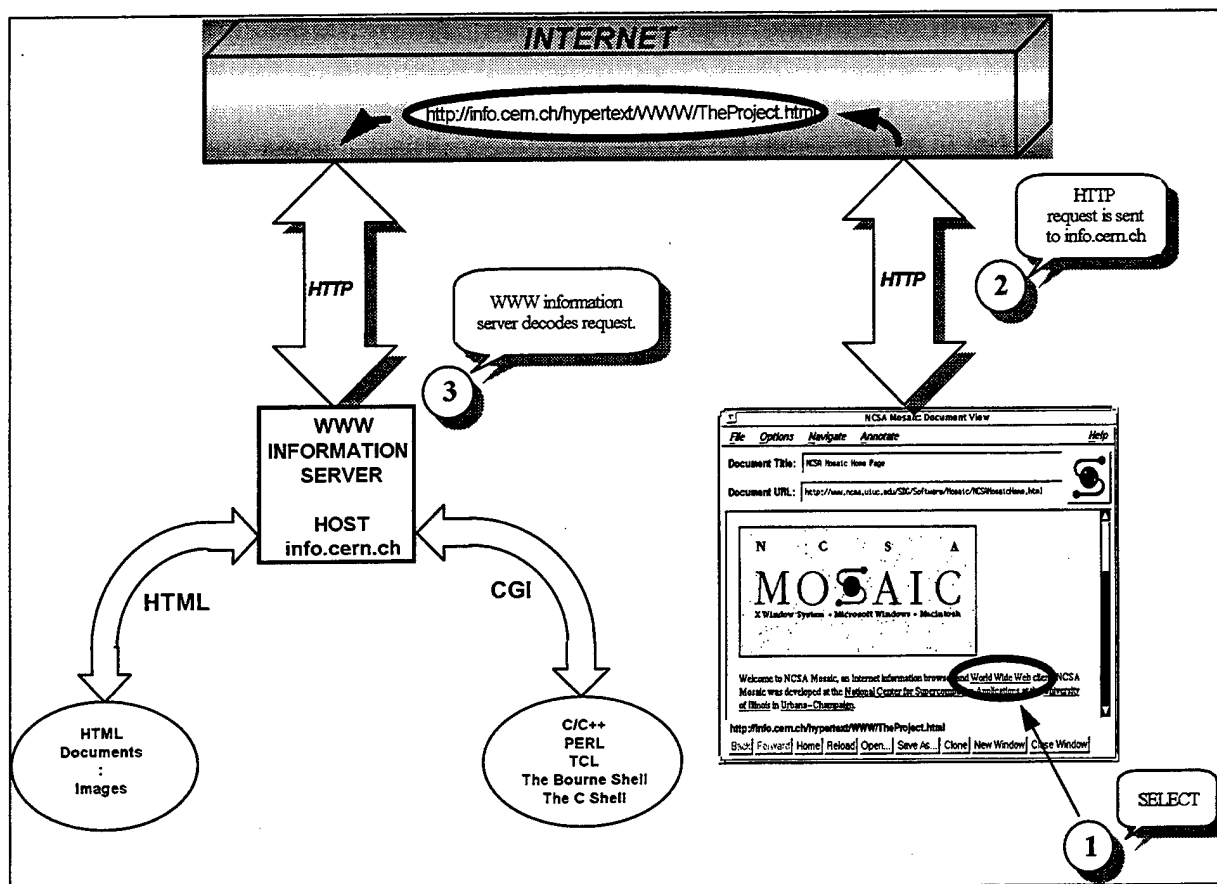
Figure 7: Hyperlink Traversal Request

Figure 7 and Figure 8 illustrate the sequence of events that occur when a user running a browser program (Mosaic in the illustration) requests information via a hypertext link. The user interface screen, shown on the lower right, displays a page of a hypertext document. The underlined phrases indicate the places in the text that are linked to other resources. The process is initiated when the user selects a link. The browser program reads the URL of the destination machine from the HTML markup that defines the link, and interprets each component of the URL: protocol, server machine name, and filename. Then the browser sends a request, in this case using the HTTP protocol, to the server machine of the destination. When these figures were developed, the hostname in the URL was info.cern.ch; it has since been changed to www.w3.org. When the WWW information server at the destination receives the request, it is decoded.

Figure 8 illustrates the return path for hyperlink requests. The WWW information server interprets the request, and builds a response. It then uses the same protocol, here HTTP, to send the resulting file back to the requesting client. Upon receipt, the client browser interprets the HTML instructions in the HTTP response to create and format the requested information on the user's display, again, as shown on the lower right of the figure.
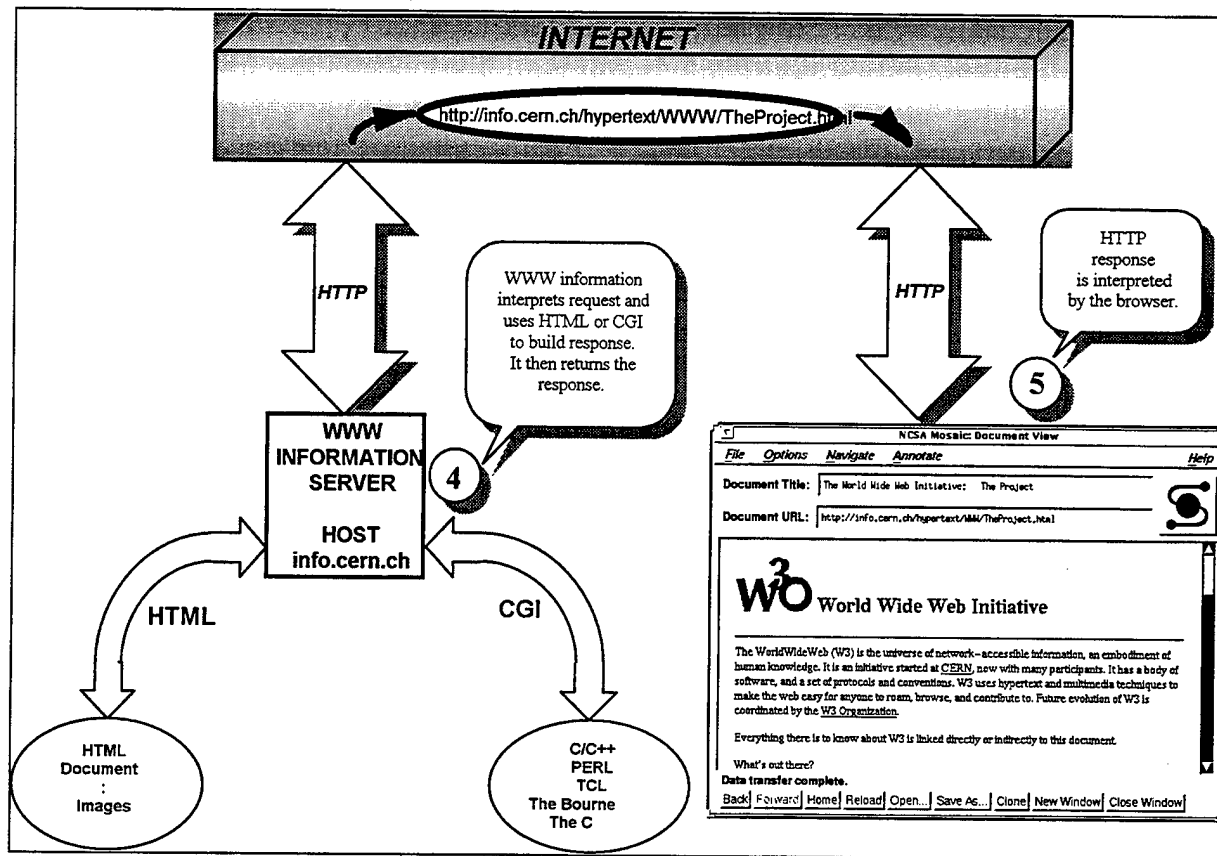
Figure 8: Hyperlink Traversal Response

2.3 Electronic Publishing and Software Engineering

Electronic publishing efforts will benefit from an engineering approach, because once information is made available as hypermedia, it must be maintained to ensure that it continues to be relevant and accurate. Throughout the handbook, issues, considerations, techniques, and procedures for electronic publishing are presented in relationship to corresponding issues, considerations, techniques, and procedures in software development. Software has been singled out as appropriate for illustrating electronic publishing because the two activities share many characteristics. Both are:

- Intangible, not able to be seen or felt
- Volatile, easily changed and rapidly changing
- Arcane, requiring special knowledge to understand

- Implemented on computer hardware
- Possible (easy) to do badly, with unforeseen long-term effects
- Products that exhibit a wide range of quality and sophistication
- Growing in popularity, becoming more pervasive
- Of unknown, and perhaps unlimited potential
- A novel way of thinking about familiar concepts

Software engineering, however, has had a few decades' head start. Web publishers can benefit from the twenty-plus year learning curve that software developers have climbed. Developments in software engineering can also be used to predict what may happen with WWW publishing. This handbook captures the WWW and HTML at a point in time, but recognizes that both are changing quickly. As a result, decisions that affect the ability to maintain quality information on the Web are emphasized. Also, the engineering guidance

10

given in the handbook transcends expected changes in technical details, such as particular tools, browser capabilities or markup language syntax.

The current state of the practice in developing documents for publishing on the Web is about where software development was in the late 1960's. People are beginning to realize that there is more to creating useful, readable, maintainable information sources than just adding HTML tags to text files and linking them together somehow. Experience in software engineering has shown that efforts spent in designing software before coding it produce payoffs in reliability, usability, maintainability, and lower overall costs. The same will be true for hypertext information.

From an understanding of how the software industry has grown and matured, other changes can be anticipated for WWW publishing. Organizations will:
- Experience growth in the volume and size of their WWW publishing activities
- Invest more and more resources in WWW-related information
- Begin to depend on the WWW to maintain their competitive positions

As a result, management issues will become intertwined with the technical ones. Managers will find that current ad hoc practices do not scale up well. They will need to find more efficient ways of achieving the same results. They will need to estimate, allocate, monitor and control the life cycle costs, including maintenance costs, of their electronic documents. Their presence on the WWW will come to be viewed as an enterprise-level asset.

Other changes will occur in the way people learn about Web publishing. Until now, the most common paradigm for learning how to write HTML or create Web pages has been by example. That is the advice given in many of the on-line HTML guides and in frequently asked question (FAQ) responses [Boutell 95]. But continued reliance on learning by example has two serious drawbacks:
- It can perpetuate less-than-ideal practices

- It is inefficient – each new user must traverse the same learning curve

These weaknesses will not be tolerated as WWW publishing becomes a larger-scale activity. The need is already being seen for training courses, and for materials (such as this handbook) which can help new users become proficient quickly, leveraging the maximum benefit from other peoples' experiences without having to repeat their mistakes.

2.3.1 Historical Parallels

The relationship between software engineering and electronic publishing can be illustrated with some specific parallels between advancements in software technology and current or anticipated developments in WWW authoring. The following evolutionary landmarks in software are discussed:
- Concern for programming style
- Availability of tools to automate software development functions
- Transition to higher order languages (HOL)
- Changes in the relationships between users and computers
- Software process improvement

Programming Style
Once software developers learned how to get programs to run, style issues became more important. Developers found they needed to get programs to be readable, so they would be understandable, testable, maintainable, and reusable. Injunctions such as "don't use GOTOs" were heard. Structured programming was invented. Rules about the ratio of comment lines to executable lines were formulated. Code formatting guidelines to enhance readability, that covered such things as indenting, capitalization, alphabetizing lists of variables, and so on, were developed. The value of selecting properly descriptive names for variables and programs was emphasized.

Style is becoming more important to Web authors now, for many of the same reasons. "Don't use GOTOs" can be translated to "provide navigation aids so users do not become lost when following a series of

hyperlinks." A GOTO is the software equivalent of a hyperlink; incomprehensibly-linked hypertext information spaces are as cryptic as spaghetti code.

Software Development Tools
The evolution of software engineering can be plotted by the evolution of tools to automate, replicate, measure, and control the software development process. Many of the newer development methodologies depend on the availability of tools to make them feasible. Rapid prototyping, for example, relies on such tools as fourth generation languages (4GLs) and interface-generators to create prototypes for users to exercise and evaluate early in the development life cycle.

The evolution of browser and development tools, similarly, is having a tremendous impact on WWW accessibility and growth. Many of the frustrations, caveats, and issues that currently plague both Web information providers and users will disappear or become moot as new generations of tools appear. Newer browsers, such as Netscape, provide users with more information about file sizes and transfer times when traversing links than earlier versions of Mosaic or any of the non-GUI browsers do.

Development and conversion tools are beginning to provide enough sophistication to shield Web document authors from direct manipulation of HTML syntax. For example, in November 1994, Interleaf, Inc. released an Internet publishing tool called Cyberleaf. This tool addresses both the conversion of existing documents and the maintenance of documents in different formats, moving towards an environment for Web authoring [Smartt 94]. The appearance of Cyberleaf and other similarly comprehensive Web publishing tools and languages, such as NaviSoft's NaviPress, Silicon Graphics' WebFORCE, and Sun Microsystems' Java, also parallels the evolution of software engineering tools, from stand-alone, single purpose tools toward comprehensive environments, such as Computer Aided Software Engineering (CASE) tools.

Higher Order Languages
Even though in theory software is written for an audience of compilers, in practice humans have to read and understand it, too. The switch from assembly level code and other unreadable instruction sets to higher order languages reflected this reality. The overall effort needed to make better compilers that could understand English-like code was less than the effort that would be needed to train enough people to understand computer-like instructions.

This same type of transition is now occurring with markup languages. The text formatting tools commonly used in UNIX environments, nroff and troff, are at best terse, whereas the rules for standard generalized markup language (SGML) tend toward more meaningful markup. Compare the descriptive tags in Table 2 with their equivalent nroff designations. Although the current definition of HTML contains both descriptive and cryptic notations, the trend is toward use of the more descriptive logical tags over use of tags that specify physical formatting.

Table 2: Readability Comparison of Markup Languages

Descriptive Markup	UNIX nroff
emphasized text	\fIitalicized text\fP
strongly emphasized text	\fBboldfaced text\fP
<blockquote>quoted text</blockquote>	.(b .ta 0.5i F quoted text .)b

Relationships with Users

As the audience for computer processing output shifted from being passive consumers of data to becoming active *users* of software, the nature of the programs themselves shifted in response. User interface design became more important, as programs needed to be more user-friendly. They also needed to be more robust, or at least fail more gently (end-users can't be expected to interpret core dumps, or abends). Software development organizations recognized and responded to these needs, providing additional services, such as customer support hotlines, training courses, and user-oriented documentation.

The WWW is experiencing a similar shift, away from a primarily scientific and academic audience toward a more heterogeneous group, including commercial and casual users. It is not reasonable to assume that readers of Web documents are HTML-literate, or even completely computer-literate. Web information providers, therefore, need to ensure that their documents are user-friendly, robust, and correct. They are also beginning to offer additional services to help their target audiences access and use the information they provide.

The population of Web authors, too, is diversifying, to include not just the scientific- and computer-oriented, but also people with backgrounds in graphics design, publishing, and none of the above. The increased sophistication of HTML development tools is helping to make Web authoring feasible for non-technical users.

Process Improvement

Software process improvement is a high-interest topic in software development organizations. Technology advances can not keep pace with the needs for increased quality and productivity. Organizations are learning that just having tools is not sufficient. The way the tools are applied, i.e., within a defined, repeatable, and accountable development process, is critical to success.

The growth in volume and importance of the WWW will soon make process as important to Web information providers as it is to software developers. Organizations that are able to define, manage, control and improve their electronic publishing processes will achieve returns on their investments that might otherwise be lost. The next section describes a framework for defining Web authoring processes.

2.3.2 Web Authoring Life Cycle

Software engineers and theorists are still debating the pros and cons of different life cycle models [Agresti 86]. The classic "waterfall" model has been criticized for not being able to accommodate prototyping, or for artificially and unproductively limiting communication between developers and end-users. Others have argued that, in reality, requirements can not be wholly separated from design. Still, the concept of a life cycle model is useful, if only to define, collect, and give names to the different types of activities that occur between the decision to begin developing a system and the decision to retire it.

A life cycle model for electronic documents, around which this handbook is structured, is shown in Figure 9. This model is derived from the management-oriented system development phases -- definition, design, implementation, and evaluation -- that were identified in conjunction with the definition of structured programming [Smith 74]. A period of definition and higher-level decision-making is followed by activities needed to transform those decisions into output. Afterward, the work involves refinement, adaptation, and update. For each of the major activity phases, associated considerations and activities relevant to Web authoring are listed on the figure. These are the topics discussed in the handbook.
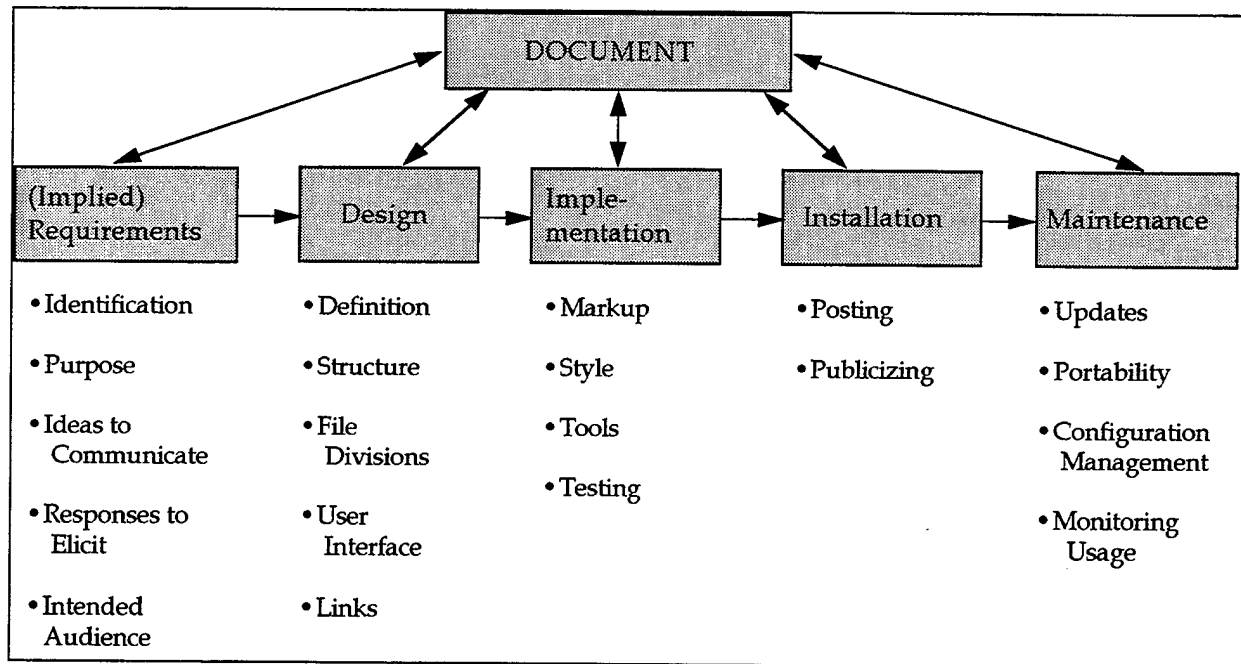
Figure 9: Life Cycle Model for Web Authoring

The continued debate about the software engineering life cycles illustrates that the appropriate model depends on the problem domain. That is also a useful lesson for Web authoring. The goal of the publishing effort, i.e., what is to be accomplished by becoming an information provider, will make some approaches better than others. This idea is incorporated in the discussion of document attributes, in Section 2.4. The real revelation, however, is that the first step must be to determine what the objectives are for publishing information resources on the Internet.

2.4 Requirements for Electronic Documents

In an engineering context, part of the design process involves making tradeoffs among competing goals. Development of an information kiosk requires design tradeoff analyses at two levels: first, for the content and organization of the overall information space; second, for the format and structure of the individual documents to be included. The tradeoff decisions depend on the attributes of the information, which are derived from its requirements. The final design will then reflect both the information content and the publishing objective of the material.

Although Web information providers do not typically prepare formal requirements specifications the way software developers do, an understanding of the requirements is needed before design decisions can be made. Some of this analysis is basic management-oriented planning work that is normally done before beginning any significant undertaking. Examples of questions that can be used to determine kiosk-level requirements include:

- What ideas are to be communicated?
- Who is the intended audience?
- What kinds of responses are expected or desired?
- What criteria will be used to evaluate the success of the publishing effort?
- What resources will be needed?

The answers to these questions will show the Web information provider what types of documents need to be created, and how the site needs to be managed. For example, if the audience is entirely internal to an organization, the contents may include proprietary information, such as employee data, that would not be appropriate for a publicly accessible service. If the development is viewed by an

14

organization as a pilot project that will be evaluated before making a permanent commitment to it, then the ability to demonstrate success, and therefore justify continuance, will be critical.

Later document design and implementation tradeoffs will be based on the type of documents to be published and their intended uses. Some documents are static, others dynamic. Some types of documents are well-defined, others are evolving. Some will be small and specific, others will be large and comprehensive.

Four categories of attributes which can be used to classify electronic documents and to guide design tradeoff decisions are listed in Table 3. Any electronic document can be characterized by where it falls on the continuum from simple to complex for each of the four categories. The table also defines the endpoints of the four attribute dimensions: extendibility, volatility, novelty and originality. The last column of the table includes an example of what needs to be considered when a document is at the complex end of the scale for any attribute.

Some examples of Web document types which exhibit different degrees of complexity for these

attributes are shown in Figure 10. The types are representative of the kinds of documents Web information providers include in their electronic kiosks. The sampling is not intended to be comprehensive, but rather to illustrate how attributes are related to document types for some easily recognized instances. Innumerable variations on the concept of "document" exist on the Web, with people adding new applications and expressions daily. In the figure, the vertical axis represents the degree of complexity, from simple (**LO**), to complex (**HI**). The complexity levels shown are not absolute; any of the document types listed could be implemented on the Web in ways that would give different profiles.

The *report* document type represents a document that is converted to HTML from an existing sequential document. This type scores low on complexity for three of the four attributes: the information is static, the concept of a report is familiar, and it can be self-contained, such that all of the links are to other components of the document. The originality score is plotted as halfway between simple and complex, to illustrate that publishing a report electronically could be done by simply converting the sequential text to sequentially-linked hypertext, or it could involve redesigning the entire document to better exploit the capabilities of hypertext.

Table 3: Design Attribute Categories

Attribute	Simple	Complex	Consideration
Extendibility	Stand Alone	External Pointers	External Link Maintenance
Volatility	Static	Dynamic	Frequency of Updating
Novelty	Users' Needs Known	Open-Ended Use	Monitor Accesses Closely
Originality	Convert Sequential	Original Hypertext	Design Structure, Layout
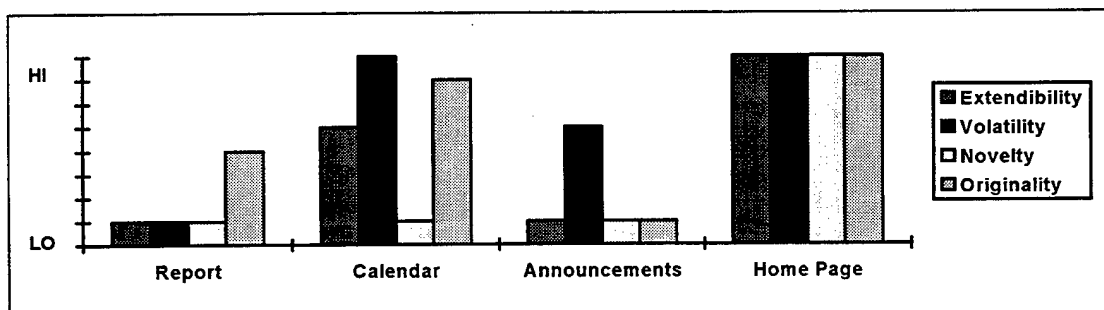
Figure 10: Attribute Complexity Levels for Selected Document Types

The document type labeled *calendar* could refer to a posting of upcoming events related to the Web publisher's area of interest, or to a schedule of activities relating to the organization itself. Such a calendar represents a type of information service that needs to be kept current, so it scores high on the volatility scale. The calendar scores low on the novelty scale because the concept of a calendar of events is familiar; many periodicals contain such listings. A useful feature of a hypertext calendar is its ability to link each listed event to a source of more information, which moves the extendibility attribute up from self-contained toward interconnected. The fourth attribute, originality, is high because this type of information is not normally maintained in paper form (it's too volatile), but is compiled as needed. Therefore, publishing a calendar on the Web would require designing and implementing it directly.

The *announcements* document represents the type of information that is typically put in a press release. The information is specific, and mostly self-contained, except perhaps for a link to the Web site's feedback mechanism. It is also a familiar type of information, and as a press release probably has a non-hypertext equivalent created and distributed conventionally at the same time. This type of document scores above the minimum for only the volatility attribute, and that is more a reflection of its immediacy and short useful lifetime than of its need to be updated.

A *home page* is the WWW term for a higher-level information page or collection of pages that serves as both a welcome mat and a main menu for a Web site, or for a lower-level collection of resources. A home page is at the complex end of the scale for each of the identified attributes: it contains links to external sites of interest, it must be updated regularly to reflect changes in the lower-level pages, how others will use it is unknown, and it has no existing sequential document counterpart. An example of part of the home page for the DACS is shown in Figure 11. It contains external links to the Information Analysis Center (IAC) program's hub page, to DTIC, and to Rome Laboratory. The DACS updates its home page regularly to reflect ongoing activities both at the DACS and in the field of software engineering.
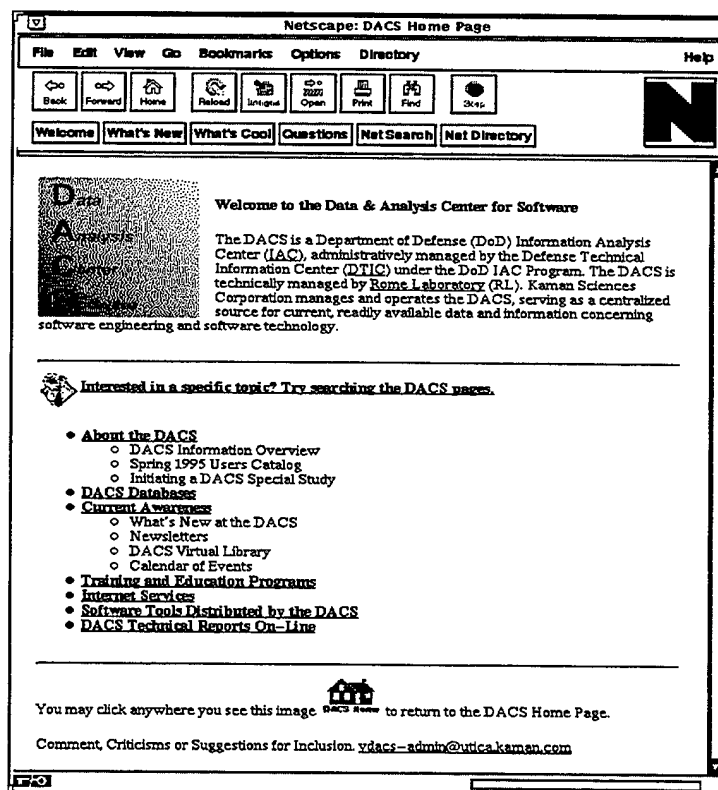
Figure 11: The DACS Home Page

3 DESIGN

Designing quality Web information spaces requires an understanding of the life cycle implications of design decisions, and knowledge of design practices which will improve the overall quality and maintainability of the information being published on the WWW. Issues to be considered in the design of documents for Web publishing are presented in terms of tradeoffs. Some of the design considerations are unique to hypertext; others are familiar concepts in a new context. Also included are specific design recommendations for some particular aspects of Web documents.

The guidance in this section is based both on WWW authoring experiences at the Data & Analysis Center for Software, and on a study of the technical literature. Further discussions of hypertext design issues, that are beyond the scope of the handbook, can be found in the sources listed in the Design section of Appendix B, including, *State of the Art Review on Hypermedia Issues and Applications* [Balasubramanian 94] and *Exploring Hypermedia Information Services for Disseminating Software Engineering Information* [Hefley 94].

An alternate starting point for beginning a design effort, although not a substitute for analysis of the effort's inherent requirements, is to examine examples of other Web sites, such as those listed in Table 4. The examples listed include both individual documents and complete information kiosks.

3.1 Design Tradeoffs

Design specifies the physical and logical structure of individual objects and the relationships among objects. For electronic publishers, design includes defining the contents and organization of the information space, and the content, appearance and structure of the individual documents and document components. Design is an activity distinct from the implementation of the components as HTML files. Web file implementation is analogous to the coding phase of software development, during which hypertext markup language tags are added to the individual files.

The idea of needing to design a document before publishing it on the Web may at first sound like something · unique to electronic publishing, but in reality documents always have a design stage. The difference is that with traditional, linear documents the designs have been so reused and so refined that they are now accepted as givens. The definition and structure of a book, for example, has been relatively static for centuries:
- Cover with title, author's name, publisher, graphics, etc.
- Cover page and its obverse
- Table of contents
- Text divided into chapters and sections
- Index

It is very familiar, but it is a design. There are variants for different types of books: novels, for example, generally don't have indexes; textbooks include review questions and exercises; mass-market paperbacks have snippets of favorable reviews on the back cover. Publishing houses employ book designers who make decisions about size, layout, and the placement of illustrations, to match each book's subject matter and marketing strategy. Web information providers need to make similar decisions about the documents they author for the Internet.

Table 4: Design Examples

SOURCE	URL (current as of August 1995)
The Best of the Web 94 awards	http://wings.buffalo.edu/contest
WWW demonstrations compiled by the National Center for Supercomputing Applications (NCSA)	http://www.ncsa.uiuc.edu/demoweb/demo.html
WWW demonstrations compiled by the Software Engineering Institute (SEI)	http://www.sei.cmu.edu/demos.html
The World Wide Web virtual library	http://www.w3.org/hypertext/DataSources/bySubject/Overview.html

The Department of Defense (DoD) has formalized this concept for documentation through the use of Data Item Descriptions (DIDs) which specify the design (content and structure) of deliverable documentation for Defense contracts. Thus any Software Design Document that has been prepared in accordance with the DID DI-MCCR-80012A will have the same outline, and will convey the same type of information about a software design. Although there are no DIDs for hypertext (yet), the concept of a specification for hypertext documents could be a useful mechanism for capturing decisions made in the design process, and recording them for later reuse.

The primary tradeoffs to be considered in making design decisions are usability, portability, and maintainability, as defined and described below. Many other tradeoff decisions can be expressed in terms of these three, often competing, goals.

3.1.1 Usability

Usability tradeoffs concern user interface design, navigation, and performance. The computer technology term *user* is the WWW equivalent of the traditional publishing terms *reader* or *audience*. The user interface for electronic documents refers to the look and feel of the pages of information. Software developers have invested much research into improving the user interfaces for their products, and a major area of concern within software engineering is the interactions between humans and computers, as indicated by the starter list of

resources on this topic in Appendix B. Lessons learned from making software applications more user-friendly, responsive, understandable, and forgiving of user-generated errors are directly applicable to improving the usability of WWW publications.

User Perspective

The Web is an appropriate name for the information space it encompasses: it is a non-hierarchical, intricately interconnected collection of resources. There are numerous, if not infinite, ways to get from one resource to another. As a result, each Web user has a unique perspective of the Web, an egocentric view of everything radiating out from the user's site as the center of the Internet universe. The user's perspective must therefore be considered in hypertext information design.

The design of any object has an implied point of view, which is the perspective from which the whole has been decomposed into its component elements. The Integration Definition (IDEF) methodology, for example, recognizes the influence of point of view on a design and therefore makes explicit whose perspective governs each level of decomposition [Mayer 92]. What hypertext and Web publishing have made more obvious is that readers have a different perspective from authors. Although this has always been true, it is routinely ignored in the design of static linear documents. With hypertext, however, users have more control over the point of view. Users can choose to read whatever bits of a document they want, in whatever order they want, without even acknowledging the existence of the rest of the

Data & Analysis Center for Software

document. Although it is not possible to predict how any user will read a hypertext document, the existence of alternate points of view can be considered in the design of documents. A further discussion of these points is contained in *World Wide Web and the Demise of the Clockwork Universe* [Munnecke 94].

Although computer-based media provide users with greater flexibility, better indexing capabilities, and more control than traditional publishing media, electronic documents lack familiar context clues such as size, positional relationships, and production values. By recognizing and responding to these differences, Web information providers can help users develop conventions for reading the new media.

Navigation
The expression "navigating the Internet" is often used, but browsing is a more appropriate metaphor because it encompasses the idea of uncertainty. Rather than steering along a predefined course in pursuit of a known goal, browsing implies looking around until something catches the attention, with no particular objective in mind. When a sales clerk asks, "May I help you find something?" the reply is, "No thanks, I'm just browsing." More experienced users have adopted the expression "net surfing." Although surfing suggests possession of particular skills and the ability to move at a greater speed, it also suggests not being in complete control, letting the Web determine both the final destination and the route taken.

As the Internet assumes a more important role in the dissemination of information, users' prevalent modes of interaction will need to shift away from unfocused browsing toward more efficient navigating. Web sites designed to capture the attention of browsing users (i.e., flashy and provocative) will be different from those engineered for navigation. Navigable sites will include the hypertext equivalent of maps, channel markers, mileposts, and other directional aids to help users stay on course. A navigable Web site will include features that function as a helpful salesclerk who guides customers to what they want quickly, or tells them that what they're seeking is not available, so they don't waste time looking for it.

Today's Web users often feel "lost in hyperspace." They suspect that much useful material may lie at the other ends of the available links, but are not given sufficient context, or indications of content, to select them appropriately. When interesting sites are found, users may not know how they arrived at them. Without bookmarking pages, or adding them to personal hotlists, they may not be able to find them again. By providing navigation aids for users, both within individual documents and across an entire site, Web information providers can increase the ability of their users to navigate through the information they publish.

For many users, the process of finding information on the Internet is an activity distinct from actually reading it. Either because they must conserve their on-line time to minimize connect-time charges, or because they prefer reading off of paper to reading off a monitor, they make heavy use of the browser's save and print functions. Inclusion of sufficient context and identifications on each separately retrievable component so that it remains understandable during later reviews is necessary to accommodate these users.

Performance
To the user, performance is synonymous with response time. The speed with which user requests are processed and results displayed affects the user's perception of site and document quality. Many of the variables which affect performance are beyond the control of an information provider: user's hardware, user's Internet connection type, current network traffic, user's browser, and supporting software. Within these constraints, however, design decisions produce a wide range of performance results. This is especially true for graphics and other non-textual information. Sometimes, just adding a warning message (or apologetic note), e.g., "This will take a long time," is sufficient. If the user knows the information is worth waiting for, the wait will be less tedious. Providing a metric that quantifies the link can be equally reassuring.

3.1.2 Portability

Portability describes the ability of a system, or a document, to adapt adequately to different environments. Some of the dimensions of WWW portability are across platforms, among browsers, and over time (i.e., newer versions of the same software). The likelihood that users will employ different Web browsers and other Internet resource discovery tools requires developers of electronic documents to consider portability in making design tradeoff decisions.

A high degree of portability is not always desirable or cost-effective. It is therefore important to determine how much portability is warranted, and along which dimensions. The answer will grow out of the requirements analysis process, where the objectives of the publishing effort are specified. For example, the WWW site developed by Silicon Graphics, Incorporated includes many graphics and multimedia components [SG 95]. The pages look and perform best on platforms that can handle the processing load, such as those manufactured by Silicon Graphics. The designers of this Web site chose to tailor it to one environment instead of making it portable - because their WWW publishing objective is more to create demand for their equipment than to disseminate information to a broad audience.

Among Platforms
Ensuring that Web documents are device-independent, and therefore portable among platforms requires consideration of what the world of readers may be using. It depends on the situation. If the Web site is entirely internal to an organization, the platform and browser combinations will be known, and may even be homogeneous. So there is no need to consider designing for other combinations (until the organization upgrades its equipment, or acquires the newer version of its browser software).

Basic differences in the underlying operating systems of common platforms can affect users' ability to access the material. For example, UNIX systems distinguish between upper and lower case, while DOS converts everything to capital letters. Mixing upper and lower case in directories and filenames, or setting up multiple URLs that differ only in case, may cause problems for users on case-insensitive platforms.

Pages which link to non-textual materials that require special viewer software may not be readable on platforms that don't support an equivalent viewer, or by users who don't have the software.

Among Browsers
Even though HTML is portable across browsers the information will not be displayed identically to users with different browsers. Even things as simple as different window or font sizes on the same platform with the same browser will make Web pages look different. More differences arise from the way various browsers format HTML markup tags, the use of non-standard HTML constructs, and browser-dependent features.

Among Users
The Web is international, which has many implications. For example, the way different cultures express dates could lead to misunderstandings: if the date August 2, 1995 is abbreviated as 8/2/95 or 8-2-95, it might be interpreted as February 8th.

Some sites accommodate user diversity by providing bilingual (and multi-lingual) page sets. From a link on the top-level page, users can select lower level pages in their preferred language.

Upward Compatibility
New browsers and Web tools may become available during the operational life of a document, and certainly will during the operational life of a site. Document design decisions that reflect limitations in browser capabilities can be reviewed, and perhaps changed, if future tools overcome those limitations. For example, the availability of interlacing has made authors feel freer to add graphics to their pages.

Awareness of pending additions to markup standards will influence designs, too. In

Data & Analysis Center for Software

recognition of the potential burden that upward incompatibility would impose on existing Web sites, those working on Web development and standards definition efforts have stated their intent to support current HTML constructs even as new versions and new markup languages evolve [Torkington 94].

Reusability
Developing a portable product may imply an intention to reuse it. In the paper documentation world, the need to control and reuse information in printed documents has driven desktop publishing systems toward more comprehensive document management systems. Document management systems deal with groups of documents. They are designed to control production, distribution, transmission, review, archiving, reuse and maintenance of documents efficiently. The concept behind document management systems is to look at the information being conveyed by the document, rather than at the documents themselves as artifacts. This is in contrast to desktop publishing, which deals with the actual production of documents, including formatting, spelling, writing, word processing, printing, etc. The approach taken by document management systems is use to Standard Generalized Markup Language (SGML) to add structure to documents, which makes them easier to retrieve and reuse [Sorensen 94]. The same needs that are driving the growth of document management systems for paper documents are applicable to electronic documents published on the Web.

The potential for reuse applies not only to the information content of hypertext documents, but also to their designs. For example, each issue of the on-line DACS newsletter (see sample newsletter in Figure 23) uses the same pattern of links to connect the pages of individual articles and sections, but the information content and the actual files are different, and reside in different subdirectories on the server. Other documents that could be designed for reuse include:

- Home page designs for a group within an organization (this is also a way to increase consistency among them, by providing a portable template page to use)

- Order, feedback and survey response forms
- Boilerplate sections of technical documents

3.1.3 Maintainability

Designing for maintainability requires an appreciation of how both the information presented and the publishing objectives may change over time. Software maintainers, for example, spend more time adding enhancements and adapting to new requirements than they do correcting errors. Maintenance of an information system is an ongoing process, that lasts as long as the system lasts, so attention to and pre-planning for maintenance activities is a cost-effective design activity.

The discussion of document attributes illustrates one dimension of the analysis required to determine document maintenance needs. Over the life of a document, its structure may change, its contents may change, or both may change. At the site level, analysis of publishing objectives and expected results will reveal the extent and frequency of maintenance effort required to achieve those goals. Details about maintaining Web sites and documents are presented in Chapter 6.

A common pitfall is trying to productize a prototype. This becomes apparent in the transition from a pilot project to a full scale publishing effort. Pilot implementations are often just proofs-of-concept, and so do not have the necessary framework on which to build an entire site: the processes and procedures used to create the prototype may not scale up; the tools may be unwieldy; and the computer resources may have insufficient capacity.

3.2 Web Site Design

The design of Web sites, or kiosks, is analogous to the preliminary design work of a software development effort, where architectural and higher-level organizational decisions are made. Web site design involves defining and organizing a collection of documents to be installed under one or more directory-type

home pages. The term home page is not precisely defined. It generally refers to the top-level page for a server, but a large site that contains many subordinate or unrelated organizations may have a home page for each distinct group. Entire Web sites may also be referred to as home pages, and so a home page may include several levels of nested and interconnected HTML files. Although an organization's home page may seem to mimic the contents of its introductory brochures or marketing flyers, there is no paper equivalent of a WWW home page.

The term home page is also used to mean a personal home page. Personal home pages are somewhat like a cross between a resume and an address book, but really have no non-Web counterpart. The Web contains many personal home pages, which vary widely from informal and personal, to formal and official. Some organizations encourage their members to create personal home pages for inclusion in their kiosks. Some even provide guidelines (for example, the DTIC Guidelines in Appendix D) describing what types of information are expected and allowed (as well as what things are not allowed). Examples of personal pages can be found all over the WWW, but they are not specifically covered further in this handbook.

3.2.1 Contents

The contents of the Web site can be determined as a result of the requirements analysis efforts. An understanding of the publishing objectives, including who the intended audience is and what the expected outcomes are, will guide the selection of information to be included. For example, a company targeting its stockholders will want to include financial data, such as quarterly and annual reports. A company interested in on-line market research will want to include registration or feedback forms to gather data on site users. The information contents of a public (external) site will be different from that served on a private (internal) system.

A starter list of contents for an organization's Web site home page might include the following types of information:
- An introduction to the organization: who they are, what they do
- Listings of products and services; available either from the Web site or conventionally
- Answers to frequently asked questions
- Avenues for feedback from Web users
- Interfaces to databases or other interactive applications
- Links to related organizations or information at other sites

Documents of different types can be included, as illustrated by the collection shown in Figure 12.

Content design includes making decisions about redundancy, such as providing different views of the same basic information. For example, an organization's press releases often report on products or events that are described in other publications, such as a company newsletter, or product information brochures. Although the basic information is redundant, the traditional audience and therefore the editorial slant are different, so the organization may want to include both on the Web site.

Another content design decision involves the expected frequency of change. Many sites schedule and advertise daily, weekly, or monthly updates, to encourage repeat traffic. The US National Park Service's Web site features a "Park of the Month" [NPS 95]. Time-Warner's Pathfinder site advertises "continuously updated" news [Pathfinder 95].

If an organization's publishing objective is to become the authoritative source on the Web for its particular specialty, then the site contents and the frequency with which they are changed will reflect that decision. Other Web sources need to be monitored continually. New sources need to be linked into a page as they appear. Outdated links need to be discarded. Alternate sources for similar information need to be identified and tracked, and decisions made on which are the best quality to be included on its own Web pages.

3.2.2 Organization

The second set of site design tradeoffs involves how the information contents of the site are to be organized. This involves grouping documents and document components into logically related sets, and defining menus of hyperlinks for navigating among them. This task is analogous to the user interface design for a software application. On a typical word processor, for example, the labels on the top-level menu bar, *File, Edit, Tools, Format, Help,* etc., represent the designer's view of the major categories of functions. The specific functions and features are found on the lower-level menus, presumably under an obvious heading. An automated telephone system, similarly, will have a top-level menu of choices, from which the caller selects the one that hopefully leads to the desired information or person.
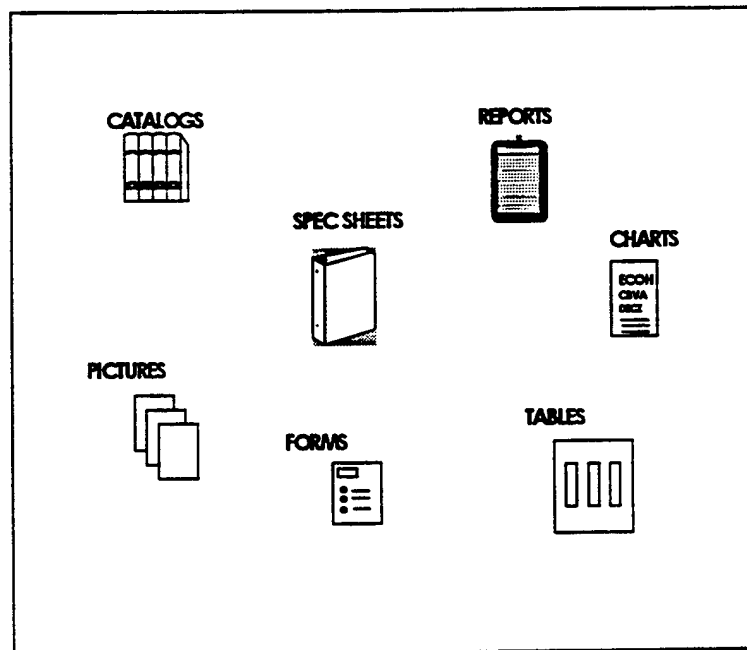
Figure 12: Generic Web Site Contents
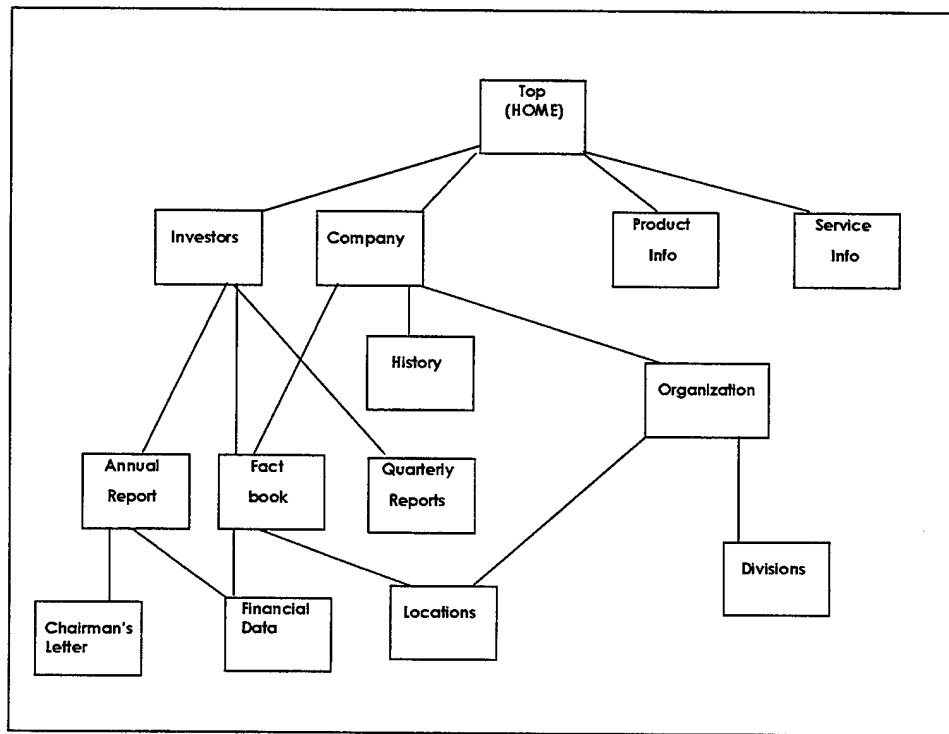
Figure 13: Sample Document Organization for a Web Site

This is a key step. Careful organization will make both navigation and maintenance easier. Factors to be considered in determining the organization are the user's perspective, the message being conveyed, and the publishing objective.

The process of designing a Web site, therefore, includes making inferences about how users are likely to browse the collection of documents. Some collections of documents, such as subject libraries, can be arranged hierarchically, with no predefined reading sequence assumed. Including cross-references and non-hierarchical links can help users can find desired material more easily. For example, the Yahoo subject library [Stanford 95] includes top level entries for both "Computers" and "Science." The entry on "Computer Science" can be reached from either one. Such cross-referencing can be overdone, however, leaving the user feeling trapped in an endless loop.

The home page for an organization could parallel the hierarchies within the organization. For instance, university home pages might be structured around schools and, within schools,

departments. A corporation might use a functional decomposition: marketing, production, customer service, research. Another may start with geographic locations, or major product categories.

Whatever the top-level categories are, they must be at a high enough level so that all the subordinate information fits logically into them. They must be robust and expandable, so that new information can be added without having to restructure the entire site. Most importantly, they must be understandable to the outside world (or the intended users, if an internal system). Common errors are to parallel the organizational structure too closely, or to label the links to lower level pages with internal jargon and organizational code names, so that only a person who is already familiar with the organizational structure can possibly know which links to follow for desired information.

One example of how a set of documents could be grouped and linked is shown in Figure 13. In the figure, links to some lower-level documents and parts of documents appear on more than one higher-level menu page,

providing users with alternate paths to the same information. This is more economical than duplicating information, as happens in paper documents, but there is some risk of confusing the user -- following an "up" link may lead "back" to a page that has not been previously visited.

Site Navigation Aids

Web authors can add navigation aids in the form of visual cues. For example, including a distinguishing icon in each individual document, analogous to the headers and footers in a paper document, will allow users to tell at a glance which document is being browsed. Such an icon would be the same throughout all pages associated with a single document, but different between documents. Logos or other identifying marks used in this way, however, need to be kept small and be constructed to also accommodate non-graphic browsers. Another common practice is to include a link to the highest-level home page of a server in each subordinate page; its disappearance immediately informs users that they have followed a link to an external site. Other useful additions are links to readme files, help pages, or feedback forms.

Navigation aids could be provided in the form of graphical displays of the current context, of navigation history, of an individual hypertext document's structure, or of the entire site's structure. Providing a graphic illustration of the subordinate pages for a top-level page would tell the user how much material is included under any given link. The page containing the site map would ideally be reachable from a link on the top-level page, rather than on the page itself, thus enabling users who know their way around to avoid waiting for a large graphic. An example of such a map is shown in Figure 14.
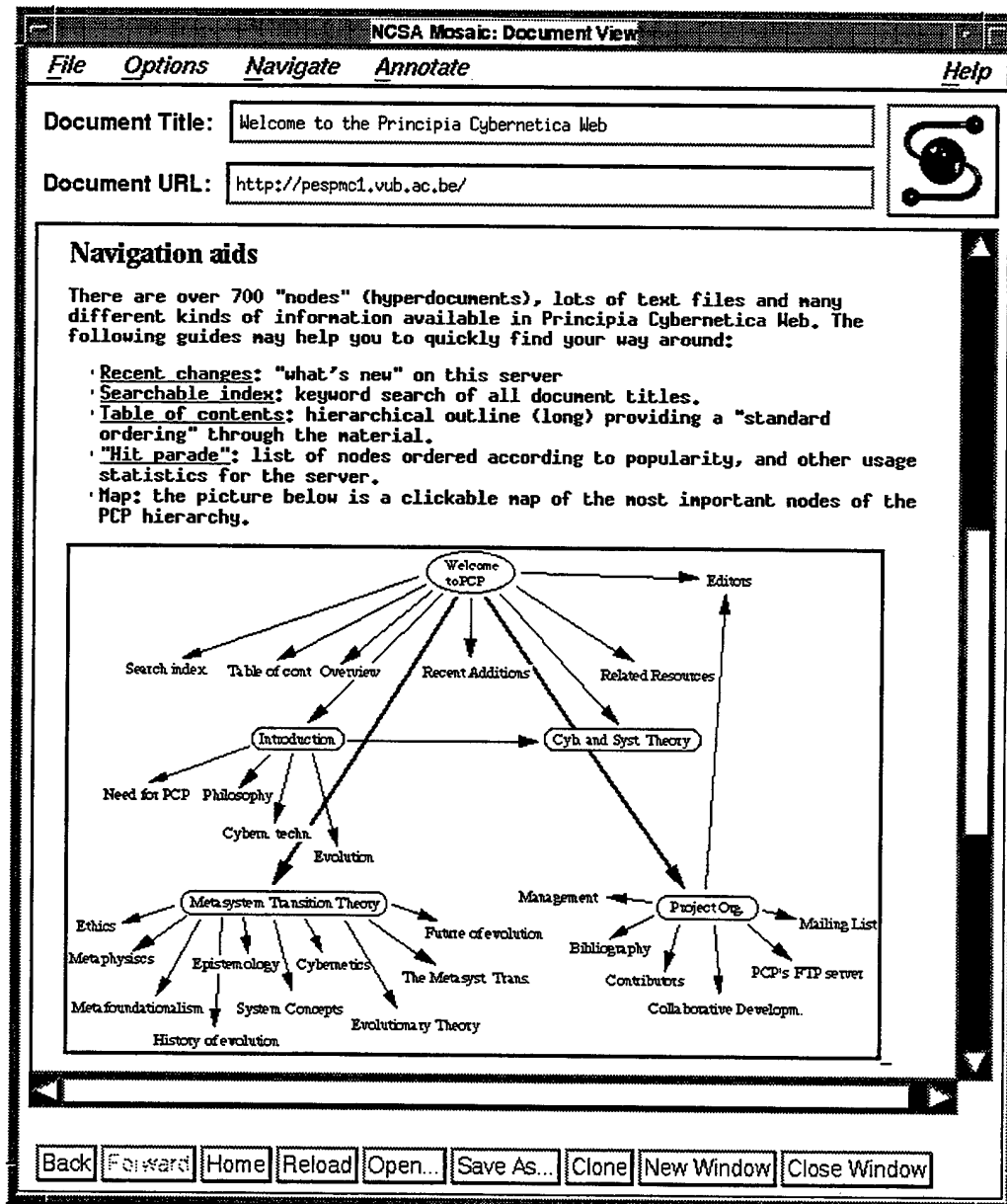
Figure 14: Web Site Map

3.2.3 Management Infrastructure

The third component of a WWW site development effort is the design of the publishing support environment. This includes: system administration; budgeting and resource allocation; assignment of ownership and responsibilities; definition of procedures, guidelines and standards to foster consistency; and incorporation of feedback and evaluation mechanisms for measuring effectiveness. The importance of these management functions can be seen from observing analogous software development efforts, where poor process management often sabotages technical accomplishments.

Webmaster
The Webmaster is the person, or group of people, who has technical and administrative responsibility for a Web site, and serves as a focal point for questions about the Web documents and information, both from within

26   Data & Analysis Center for Software

the organization and from outside users. The Webmaster's contribution to the success of a publishing effort is frequently overlooked in WWW literature.

The Webmaster function is analogous to that of system administrator for a computer system or network. Other functions may include: providing formal or informal HTML and Web training for users within the organization; controlling access permissions to server files; implementing and overseeing quality control and configuration management procedures for Web documents; and maintaining HTML files after they are published. A first step in developing a Web site, therefore, is to define and staff the local Webmaster function. As long as the Web site remains active, the Webmaster requires continued support.

Resources

A Web site needs to have sufficient resources allocated to it. In addition to the initial acquisition and set-up costs for hardware and software, the telecommunications connection, ancillary equipment such as security devices, and the costs to design and implement the Web information products, the site will consume resources as long as it remains operational. Some side-effect costs of establishing a Web presence are:

- Maintenance of the published information - document additions, updates and replacements
- Site administration - Webmaster functions
- Collection and interpretation of usage data
- Staffing and procedures for responding to inquiries generated through the Web
- Training of internal users
- Productivity effects on the information provider's own organization

Ownership and Responsibilities

A related issue involves determination of ownership of the WWW site within an organization, and specification of the resulting responsibilities. For example, responsibility for an internal-use-only system might be distributable among all the groups which contribute to it. The owner of an external system, however, needs to have the proper perspective, such as would be found in a public relations department.

Consistency and Guidelines

A software engineering truism is that consistency increases both maintainability and understandability [Booch 86]. For WWW information spaces, where effective communication and ease of maintenance are primary goals, consistency is equally desirable. One mechanism for increasing consistency across a Web site is to develop design and implementation guidelines for Web authors. Many organizations have developed style and content guidelines for their Web pages. Rome Laboratory (RL), the Air Force's software technology research laboratory at Griffiss AFB, set up a committee to design a common look and feel for RL documents on the Web [RL 95]. The Defense Technical Information Center (DTIC) requested that the IACs they sponsor include links to both the DTIC hub page and to the IAC Directory page on their home pages, to create at least a minimum level of consistency among them [Lyman 94]. Appendix D contains a more detailed guidelines document developed at DTIC for DTIC Web authors. The Government Information Locator Service (GILS), described in Section 7.4.2, is similarly aimed at improving the usability of Government-sponsored Web sites by specifying a consistent set of identifying information to be included in Government sites.

Essentially, design guidelines provide a means to increase consistency by placing some restraints on creativity and flexibility within an organization, assuming the reason for setting up a site is to convey information. A secondary result of adhering to design guidelines is to refrain from putting features into electronic documents solely because it is technically possible to do so. This is analogous to the plague of multiple fonts, also known as "ransom note typography" [Sorensen 92], practiced by novice desktop publishers who feel compelled to use all the capabilities of their powerful tools. The problem with including irrelevant technical wizardry is that it can obscure the information content of the documents.

Development of and adherence to design guidelines will help Web authors maintain consistency in their Web pages over time and among different authors. It is often possible to "date" documents and groups of pages at a site by means of subtle changes in design -- layout, structure, navigation aids, etc. Similarly, Web authors may display distinct personal styles in creating Web pages that an organization may or may not want to permit.

Design guidelines can also provide welcome guidance within an organization for those who want to create quality pages, but do not know how. One Internet user's response to a request for advice on how to develop a home page included the following, "some of the worst HTML I have seen was written by technically adept folks who had no idea of structure or style. Some of the HTML out there brings a new meaning to the term 'content-free' (especially when the page begins with a huge and meaningless graphic image... ick)" [Schneider 94].

Feedback and Evaluation
The Web site management environment needs to include some means of measuring both the use of and the users' perceptions of the site. Collection and analysis of access data and other feedback enables a Web information provider to evaluate how well a Web site is meeting its publishing objectives. Feedback can be used to fine-tune the design of a Web site, to make the information better and more useful.

Feedback on information quality and usefulness includes both active and passive information gathering. Passive information gathering is accomplished through the use of server logs and monitoring tools. Such access monitoring provides information about the user audience, and may reveal needs for links to other parts of the Web, or to other documents on the server. Tracking accesses to individual pages shows how readers trace through documents. With old-fashioned paper documents, there is no way of knowing whether or not readers find the organization and presentation useful, confusing, or boring. With electronic documents, however, the Webmaster can "watch" how people read them. Tracking the traversal

patterns of individual documents can give clues as to what information readers find important, or how it might be structured for different types of users. One can see whether people tend to come "in" at the "beginning," and if not, make sure that the pages they do access provide sufficient information to identify the document and its context.

Active feedback can be solicited from users by adding a link to an E-mail address which makes it easy for readers to provide on-the-spot comments (similar in concept to a reply card in a magazine; but one that doesn't require a pen, postage and a trip to the mailbox, or fall on the floor when picking up the magazine).

The number of statistics gathered and the amount of time spent analyzing the data depends on how the data will be used. For example, the DACS and other IACs are contractually required to report inquiries and distributions of products, as part of their current awareness programs. Therefore, accesses to the DACS home page, and to each of the products available on-line, need to be recorded for reporting purposes.

For individual documents, the amount of monitoring to do depends on the type of information being published. A home page or a reference-type resource will require closer scrutiny than something like a technical report or a conference paper, which remains fairly static once it is published. The amount of analysis also depends on the stage of an individual document's life cycle. Comprehensive statistics on page to page access would be useful during its initial period of availability on the Web, so that user reactions and traversal patterns could be determined, and the document modified to correct any deficiencies that become apparent. After an initial trial period, however, an overall count of document accesses and/or list of users may be sufficient. For a document that is published periodically, like the DACS newsletter, each page might be monitored closely for the first few issues. Then, based on the feedback and usage statistics, the developer can decide on the best design for newsletters. Subsequent issues

incorporating those design decisions would not need continued detailed low-level monitoring.

3.3 Document and Page Design

Document and page design are analogous to the detailed design tasks of a software development effort: the data and algorithms are defined for each component, and the interfaces between them are specified. By the end of the detailed design activity, all remaining decisions are about implementation, i.e., how to do the actual coding. Web document design includes defining the content, appearance and organization of each page, as well as determining the placement of internal and external hyperlinks that connect the individual pages to each other and to other Web resources.

A WWW document is an object composed of one or more separately retrievable pieces of information that are connected among themselves and possibly to other documents by hypertext links. Each separately retrievable piece of a document corresponds to a separate file on the Web server, and so has a distinct filename that is part of its URL. The term "page" has many definitions in the context of the WWW, ranging from a single screenful of information to an entire Web site (as in one interpretation of "home page"). The definition used in this handbook equates a page to a file, making pages the basic unit of information on the Web.

3.3.1 Document Structure

Document structure design tasks consist of decomposing documents into separate files and defining the hyperlinks that reconnect them. The decomposition into files affects performance. The pattern of inter- and intra-document hyperlinks affects the user's ability to navigate through and understand the documents.

Hypertext allows designers to consolidate common information, and point to it from wherever else it is relevant, instead of having to repeat the same information in multiple documents. But hypertext also allows users to read documents in piecemeal fashion. Although Web document designers may think of a collection of pages as a single document, others may find some subset of these pages more interesting than the remainder, and so include pointers into only selected pages of the document. As a result, the links from others' pages may be to pages that were originally designed to be somewhere in the middle of a larger document. The risk to the information provider is that users will miss important information by not following one or more links. Designers therefore need to provide enough information on each page so that users can understand it without having read any particular previous page. The tradeoff is between making each file completely self-contained, and making a set of files overly redundant. Although it is the user's responsibility to follow links to important information, it is the designer's responsibility to provide the links, and to label them adequately.

Page Size
Deciding how large to make each Web page is a key performance tradeoff. Because each page is implemented as a separate HTML file and is therefore retrieved separately (along with all its inlined graphics), the number and sizes of the files in a document will influence users' perceptions of its quality and its usefulness. The larger the file, the longer it takes to be retrieved over the Internet.

The decomposition of documents into pages is analogous to the decomposition of software into modules. There is no one best answer. Fifty lines of code was a long-quoted rule of thumb for module size, but the best size for any particular module depends on its function, its language, and its environment. Similarly, two screen-length pages has been suggested as proper for Web files, but the appropriate page size depends on the characteristics of the document. A completely unachievable goal that has been suggested for determining page sizes, because there are too many uncontrollable variables, would be that it takes at least as long to read a page as it does to retrieve it.

Another variable to contend with is that users' screen sizes vary. Depending on what platform a browser client is running, one screenful might hold only 20 lines of text (e.g., on a VT100 terminal), or it could hold 80 or more lines. Users in windowing environments also have the ability to resize their browser screen displays as they see fit. User-adjustable font sizes, which many browser/platform combinations allow, will also affect the display size of a page.

Assumptions about how a document will be read affect the page size decision. A designer may put a document that is likely to always be read in its entirety into one big file, so the reader will only have one (long) wait to retrieve the information. If the document is very large, however, the Web author might provide only an abstract and/or table of contents on the Web in HTML, and make the complete document available via some other Internet tool, such as FTP or E-mail.

A document such as a newsletter or electronic journal that is made up of logically separate components would be practical to decompose into separate files; one (or more) per article. This is based on the assumption that readers won't necessarily have the same level of interest in every article, and will retrieve only the parts they intend to read.

Navigation Links
Once a document has been decomposed into separate files, the designer needs to specify hyperlinks to connect them so that users can navigate the document. For individual documents, navigation aids can include:

- A table of contents page, linked to each section (see Figure 15)
- A searchable index that provides a user with alternate entry points into the text
- A graph or diagram of the pages
- Information about the size and contents of the files that make up the document (see Figure 17)

For Web documents that do not have a corresponding paper equivalent that a user can conceptualize, ensuring that document hyperlinks reflect rather than obscure the document structure is even more important. The arrangement and naming of a document's component files into directories and subdirectories can also be used to help convey the document's structure to users, because the filename and pathname are included as part of each page's URL.

Many Web documents are structured like traditional linear documents. A table of contents on the top-level page of such a document, with section headings serving as links to the other pages of the document, is shown in Figure 15.

Data & Analysis Center for Software
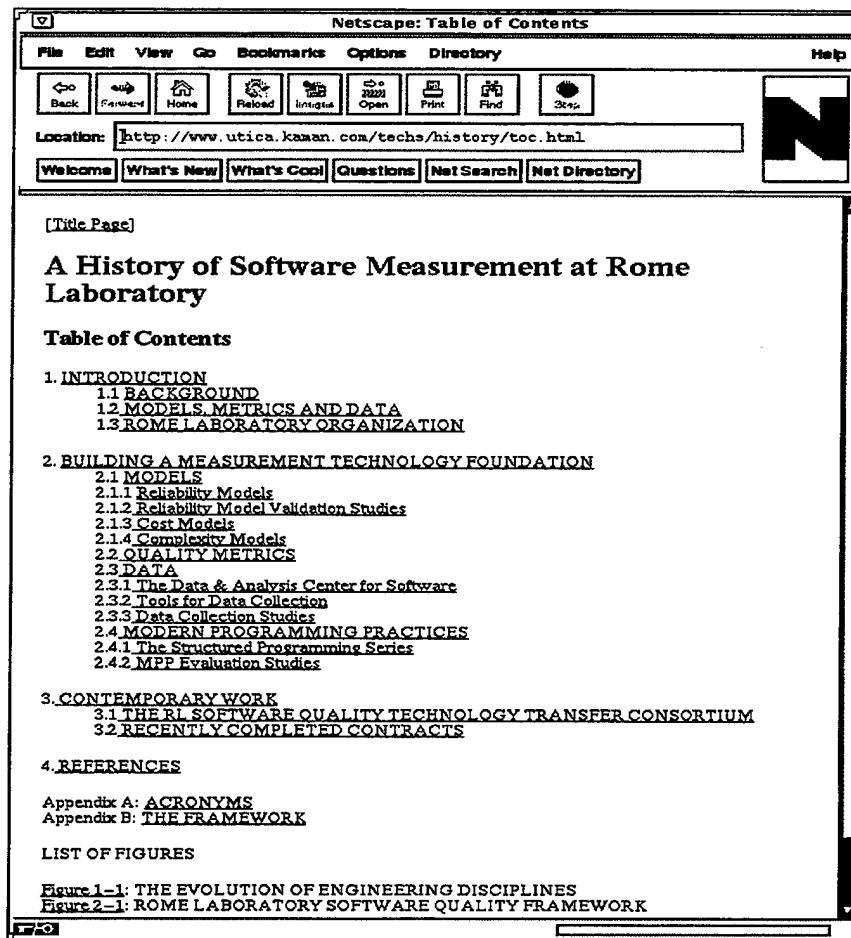
Figure 15: A Table of Contents Page

Alternate reading sequences could be defined for the same document by providing multiple tables of contents. For example, this handbook has been written to serve different audiences, as described in Section 1.2. In the hypertext implementation, each view of the material in the handbook is represented by a separate, tailored table of contents, reflecting the suggested relevant sections as depicted in Figure 1 through Figure 3.

Table 5: Link Equivalents

Link	UNIX	DOS
Top	*chdir*	*c:*
Up	*chdir ..*	*cd ..*

For individual pages that are part of a larger document, providing links (labeled "Top" and "Up") to the title page and highest-level page for that document will allow users to quickly get a higher-level view of the material, to reorient themselves, or to return to a previous location without retracing a long sequence of hyperlinks. The analogous computer file and directory navigation commands are shown in Table 5.

A distinction needs to be made between the static sequence of links associated with a document and the dynamic sequence of links associated with a user's series of accesses to it. Many browsers include "back" and "forward" buttons or commands as part of their user interfaces. Including document-specific "back" and "forward" links can be confusing, and is
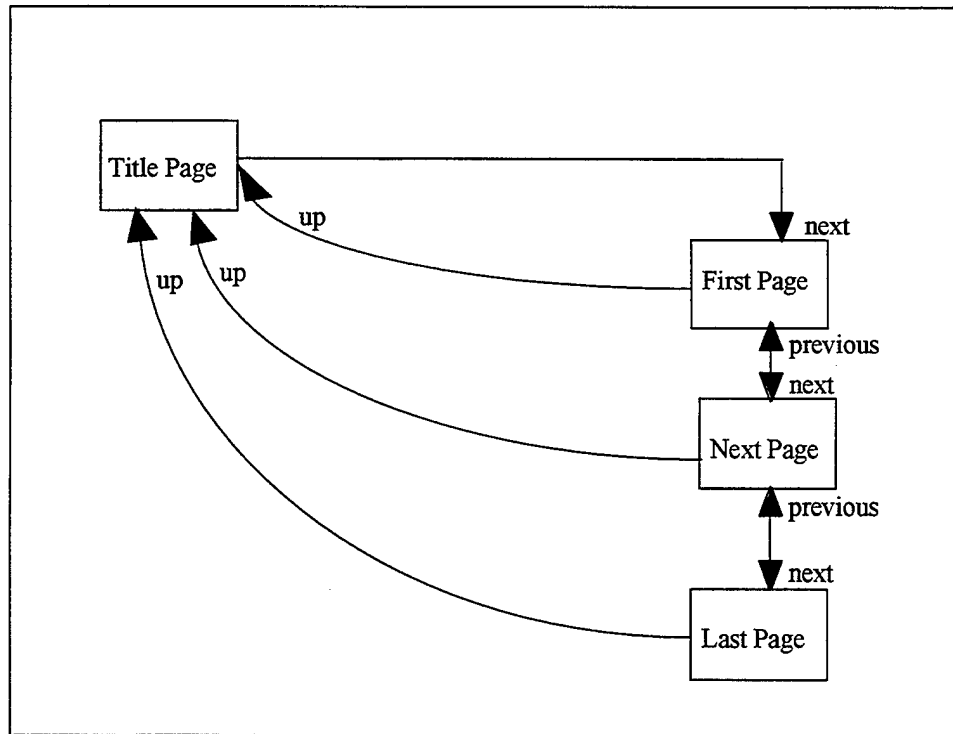
Figure 16 : Navigation Links within a Document

therefore not generally recommended for hypertext. If a particular sequence is important to the understanding of a document, however, the pieces can be explicitly linked together in order using "previous" and "next" to identify the document-specific sequence, leaving "back" and "forward" to describe the user's path. Some conversion tools create such document-specific links automatically. A graphic illustration of navigation links among document components is presented in Figure 16.

Another way to increase the ability of browsing users to find and use the information they seek is to add information to the hypertext links. By making them more descriptive, users will have a better idea of whether or not following a link

will provide the expected information, without having to execute it.

Two types of additional information can be provided. The first is a verbal description of the material at the other end of the link, and may include the author's assessment of its quality or relevance. This is the most commonly used approach, and is the easiest to implement. The second type of additional information provides technical or implementation details from which a user can infer the performance impact of selecting the link. This information could include the size of the linked file in kilobytes of text, the number of images it contains, and the number of links it has to other files. A prototype example of how this could be done is shown in Figure 17.
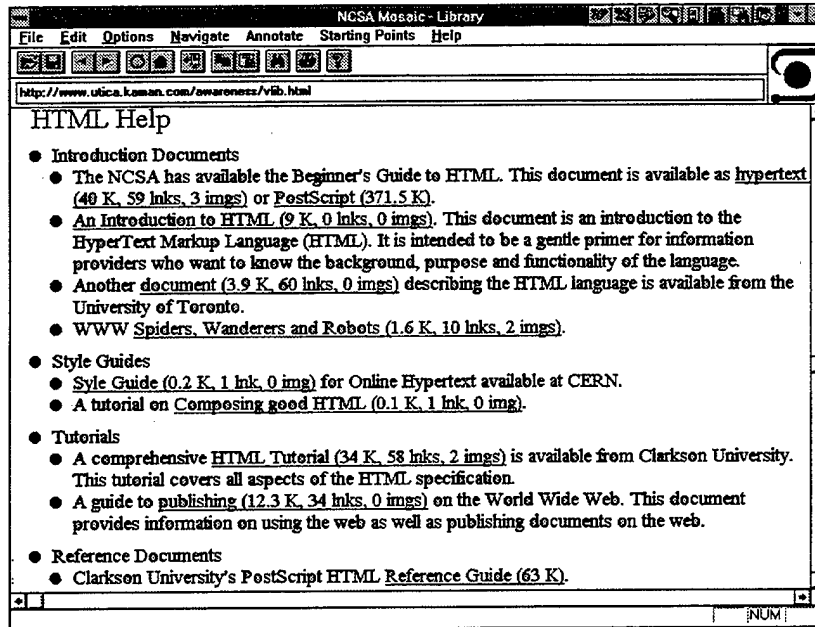
Figure 17: Descriptive Links

Although the second method is more thorough, it is also more difficult to maintain. It is really only feasible to consider for files being hosted on a local server, and then only for information that is relatively stable. For external links, providing this information and keeping it current would require a much larger investment in maintenance effort than could be justified by the value added to the page.

Comprehension Links

Hyperlinks within the text of a document are used for more than just navigation; they can also add usability to hypertext documents. For example, the standard practice in technical writing is to expand an acronym the first time it is used in a document, e.g., the Data & Analysis Center for Software (DACS). From then on, only the acronym, DACS, is required. The reader is expected to remember what it stands for while reading the remainder of the document. A collected list of acronyms will usually be provided at either the beginning of the document or as an appendix. Readers can keep a finger inserted in this list in order to flip there easily every time a forgotten or unfamiliar acronym is encountered in the text.

With non-sequential hypertext documents, however, the location of the "first appearance"

of an acronym is undefinable. One solution to this problem would be to expand each acronym the first time it is used on each separately retrievable page of a document. Another solution, that exploits the characteristics of hypertext rather than fighting them, would be to provide a hyperlink between every acronym used in the document and its expansion/definition on a separate acronyms page or section. If the number of acronyms in a document is large, the list could be split into several pages, so each can be retrieved quickly.

Hyperlinks can be created for any reference portions of the document that are provided to help a reader better understand the material. Essentially, any part of a paper document that a reader would tab or earmark for easier reference is a candidate for linking into the text. Examples of these that are typically found in technical documents include:

- Lists of abbreviations and acronym expansions
- Units of measure, or conversion formulas
- Glossaries
- Diagrams
- Explanatory appendices
- Trademarked words
- Foreign words and phrases

- Footnotes (a footnote tag has been proposed for HTML3)

3.3.2 Page Layout

Determining the content and format of individual Web pages requires more tradeoffs. Web page design is analogous to the user interface design of a software system; Web pages are what the user actually sees and interacts with. Design choices in the following areas affect the performance, readability, portability, and usability of Web documents:

- Inclusion of graphics: informational, icons, decorative, identifying
- Logical formatting (rather than physical): use of headings and lists
- Consistency: use of page templates
- Common links: gohome, mailto, help, copyright notice, title page

One performance tradeoff is between the layout overhead and the information content of each page. Although it is useful (and recommended) to include identification, navigation, and other graphic objects on Web pages, the performance penalty they add to a user's retrieval of the page must be kept relatively small in proportion to the actual information they surround. Users will be frustrated after waiting for a page full of large banners, imagemaps, header and footer graphics, or intricately crafted navigation button bars to load, if they find it only contains a sentence or two of text, or a few unannotated links to more pages.

Logical formatting means that the relative positions of items on a page are determined by their logical relationships, e.g., subheadings are subordinate to headings. Because each browser interprets HTML differently, it is not possible to specify exactly where on a page each component will be displayed. More tags will be available in the next version of HTML to specify formatting and placement more explicitly, and some browsers already provide support for formatting instructions such as centering, tables, and text-wrapping around images. For now, however, if pages are to be portable among browsers, designers must rely on

content and logical relationships to present their messages coherently.

The ultimate display layout and format of each Web page is under the control of the user and the user's browser. HTML was designed that way on purpose, to provide more flexibility, and to be more universal. The user and/or browser can select the display screen size and dimensions, the foreground and background colors, and the typeface and point size. Many Web page designers, especially those with backgrounds in graphics design, have difficulty relinquishing control over the presentation of their works.

Development of a standard page layout provides a useful way to promulgate design tradeoff decisions across a set of documents, as well as to increase consistency among them. An example is shown in Figure 18. The template specifies the size and relative location of graphics, identification conventions, and standard navigation, feedback or help links. The template may be specified for only higher-level pages of long or complex documents. Different templates may be created for different types of pages and documents. Some HTML authoring tools and services provide page templates (which incorporate the tool developers' design decisions) into which Web authors can insert their own contents.

Document Information
Most graphical browsers display the title of each page, and can be configured to also display the URL of the current page. Even assuming that serious users will display the URLs of the pages they access, the title of each HTML page is also important, because it is usually titles, not URLs, that are saved in the browser history.

When browsing HTML documents on the Web, location information is embedded in the link that connects the user to the site. When HTML files are printed or saved as PostScript or ASCII files, however, the URL information is often lost. Because many users employ a "surf now, read later" approach, or print out Web pages for further distribution, it would be useful to have the WWW location information saved

automatically with the files. One way to do this would be to manually include the URL on each page. This concept could be expanded to include an extended set of pro-forma document information at the bottom of the page, as illustrated in Figure 19.

Since users will not be interested in this information unless they are also interested in the document, it must not consume a great deal of time. A hyperlink at the very beginning of the HTML page, preceding any text, will point to this reference information at the bottom of the file. If and when the page is printed or saved as a file this information would provide enough detail about its location for easy access and retrieval at a later time. Values for some of the fields, such as date last accessed, could be set automatically by the server. The knowledge prerequisite field can be used to classify the usefulness of the document relative to the user's subject knowledge. "None" indicates the user needs no previous knowledge to understand the document. This may also imply that it is preliminary information which advanced users may choose to ignore. "Familiar" would require some basic knowledge, similar to being familiar enough with the Web to have gotten to that page. "Advanced" would indicate that the author's intended audience for the document has significant experience in the subject area.
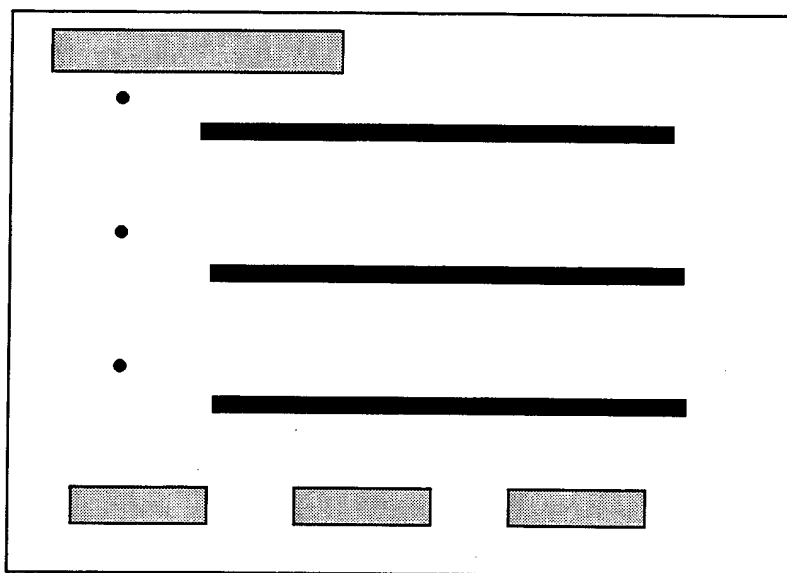
Figure 18: A Page Layout Template

```
TITLE: The Title of the Document

                    Extended Document Information
                          (linked to bottom of the page)


    Text of document.....




               (BODY OF DOCUMENT / HTML PAGE )




--------------------------------------------
extended information area
```

```
Extended Document Information

                  Title: "The Title of the Document"
           Access date: indicates the last time someone viewed the file
                    URL: <complete pathname>
                 Author:  <author's name>
   Author e-mail address: <author@site>
              File DATE: indicates the last time the file was changed
            File Format: <HTML> <ASCII> <PostScript> <Other>
              File size: <nnn K>
          Organization: <document owner>
               FTP site: <site> <none>
      Alternate Formats: <ASCII> <PostScript> <Other> (with URL)
 Knowledge Prerequisite: <None> <Familiar> <Advanced>
 WEB site administrator:  <name and e-mail address>
```

Figure 19: Sample Extended Document Information

3.3.3 References in Hypertext

A unique characteristic of hypermedia is the ability to include links to the actual sources of information in the documents. In a paper book or journal, footnotes and references can only identify the sources of quoted or related information. Searching them out requires a second trip to the library, and good research skills to find the document or to determine where a physical copy is located. If the source is in a reference work or bound journal, one then needs change for the copier. If it's not in the local library's holdings, one needs the patience to wait for an inter-library loan request to be fulfilled.

In a hypertext world, the distinction between original and source material becomes less obvious. By following links, the readers can explore reference and supporting material while they are reading the document. Access to

Data & Analysis Center for Software

sources is immediate, relative to the paper library approach. Access is not instantaneous, however, and can seem unacceptably slow relative to users' accustomed computer response times.

The tradeoffs are in performance and maintainability. Within the text of a document, citations are not linked directly to the source, but to an intermediate footnote, reference, or bibliography page. Links to the source documents themselves are instead provided on the reference page. This gives the reader the chance to get more information about the source, before deciding whether or not to "send for" the actual document and incur the wait required for it to be retrieved and displayed. As illustrated in Figure 31, this approach increases the maintainability of the electronic document, because the links to other documents are defined in a single place. As changes occur to the referenced information, the link information can be found and updated more easily.

Referencing vs. Incorporating
Designing a document that depends on outside resources for completeness requires making decisions about whether to actually copy the material into a document or just to include a link to where it resides on the Web. The answer depends on several factors, which include characteristics of not only the document being authored, but also characteristics of the external resource and the intended audience. Things to consider include:
- Size of the external resource
- Type of external file, e.g., text, image, sound
- Compatibility of external file format with users' browsers
- Stability of the external resource
- Stability of the document of interest
- Criticality of the external information to understanding the material
- Known copyright restrictions on the external resource
- The locations of the readers with respect to both the Web site and the external resource

The last consideration is included on this list as a reminder that what is faster at the author's Web site may not be faster for distant users.

For an internal kiosk, however, the users' response times will be the same as the author's.

There is also a question about whether or not it is necessary to notify the owner or provider of an external resource before adding a link from a document to that external resource. Some Internet users consider it impolite to link without permission, because of the potential for overloading servers at popular sites by adding other paths to them [Wiggins 94].

Acknowledgments
It requires extra effort to be disciplined about acknowledgments and credits in Web documents. Quotes from informal sources, such as E-mail, newsgroups or listserver discussions often provide excellent material, but there is not yet a well-established method for citing them. There is a temptation toward sloppiness: why take the effort to carefully footnote a source, when the source is ephemeral, and will be gone before any reader ever sees the citation?

Other questions arise from the non-discrete nature of hypertext documents. It is not yet clear how to define the boundaries of a Web document. Is it appropriate to use the same citation for anything found on the same server? (Probably not.) In the same directory, or subdirectory? (Possibly.) Or must every separately retrievable page be cited separately? (Hopefully not!) If a hypertext document has been structured to provide different views of the same set of pages, is each view a unique source, and thus deserving of a separate citation?

How can a source be fully referenced when the pages of it have no identifying information on them? Users may think they are reading one document, based on the path traversed to find it, when in reality, some link has taken them to a different source entirely. Most browsers can display the URL of each page that they retrieve, but relying on URLs to identify sources can not be a complete solution because they are so volatile. An organization that is concerned about intellectual property rights is likely to be careful about identifying itself and its copyright restrictions on its WWW pages. Although such

Referenced WWW Source
[Berners-Lee 94c] Tim Berners-Lee, *Style Guide for On-line Hypertext*, at URL http://www.w3.org/hypertext/WWW/Provider/Style/Overview.html, Cited October 24, 1994; 1642 bytes.
Unreferenced WWW Source
Nathan Torkington, *Primer on WWW Servers*, at URL http://www.vuw.ac.nz/non-local/gnat/www-servers.html, updated November 19, 1993, 11658 bytes.
Referenced Newsgroup Posting
[Smith 94] Frazer Smith, fsmith@fox.nstn.ns.ca, *PowerPoint to HTML*, Article 777 of comp.infosystems.www.providers, June 30, 1994.

Figure 20: Internet Citation Formats

notices are not required for material to be protected, their inclusion or omission can be used as a rule-of-thumb measure of how the organization expects its material to be cited.

Citation Formats

There are many formats for citations within the text of a document; some common ones are shown in Table 6. The format typically used in DACS documents is to indicate the last name of the principal author followed by a two-digit year of publication in parentheses or square brackets after the quoted or referenced material, e.g., [Wolf 94]. If the same author has more than one cited publication in a single year, they are distinguished with letters, e.g., [Berners-Lee 94a], [Berners-Lee 94b]. Hyperlinks can be created in the document to point from where the reference is mentioned directly to its corresponding entry on the bibliography page.

Formats for citations on the bibliography page depend on the type of material, and on whether or not it is referenced elsewhere in the text. Citations of traditional sources use a standard bibliographic format, such as is specified in the

Table 6: Sample Citation Styles

Example	Description
[1], (1)	sequential numbering
[8, 13]	non-sequential numbering
[Samuelson 94]	author, year
(SMIT 94)	abbreviated author, year
(TOOLS)	mnemonic title or subject

National Library of Medicine (NLM) Recommended Formats for Bibliographic Citation [Patrias 91]. For items referenced in the text, the bibliography page will include the citation tag that was used in the document.

There is no standard format for citing Internet sources. The NLM manual includes bibliographic formats for other electronic media, including on-line databases, bulletin boards, CD-ROMs and videotapes, which can be used as guidelines. At a minimum, a WWW citation will include the URL where it was found. Title and author will be available from any reasonably well-constructed source. E-mail or newsgroup postings usually have sufficient information in the header portion of the message. Instead of a publication date, the "date last updated" on the source document can be used. If no date is given on the document, the date the link was traversed to access the source could be used, e.g., "Cited Aug. 20, 1995." Some browsers display the number of bytes in a document as it is being retrieved. This value could be used in the citation in lieu of a page count. If the document has embedded image files, that information could be included, too, e.g., "12508 bytes, plus images". Some examples are shown in Figure 20; other examples can be found in the references for this handbook, in Appendix B.1.

3.3.4 Converting Documents

A large percentage of the information published on the Web is converted from existing linear documents, rather than created as original hypertext. Just like the conversion of a character-based software application to a GUI environment can be accomplished by merely

changing the function keys to on-screen buttons, an existing document can be converted to hypertext by simply adding HTML tags. The results, however, will be neither true GUI software nor true hypertext, because they haven't been redesigned to take advantage of GUI or hypertext capabilities. Critics of such hastily-produced hypertext and similarly created CD-ROM products refer to these documents as "shovelware." The effort needed to effectively complete the conversion of a large linear document into a useful web of hyperlinked components can be greater than the effort required to create a hypertext document from scratch, because of the work needed to identify relationships within the text [Balasubramanian 94].

For example, converting from a sequential document to hypertext allows the designer to consider adding a second or third point of view to the text, but the designer needs to understand what the different points of view are, as well as the reasons for including them. Converting a document to HTML provides the designer with an opportunity to add multimedia components to enhance the presentation of the material in the document, but the designer needs to be sure the additions are truly enhancements, and not annoying clutter.

How much redesign is to be done depends on the requirements of the situation. If the goal is just to get a lot of existing material available to Web users quickly, a straight conversion is appropriate. If, however, existing documents are being converted as part of a complete Web site development, then the extra time invested in redesigning them as hypertext, and making them look consistent, is justified.

The selection of a conversion tool (including choosing not to use one) will affect the number of design decisions that need to be made. The DACS conversion experiences documented in Appendix C illustrate some of these differences. For example, one tool may break a document up into separate files, and link them together automatically, based on levels of headings, or some other criterion. Another tool may only add HTML tags to individual files, and thus

require the designer to have partitioned the document and identified intra-document links before using the conversion tool.

The choice of conversion tool can also affect the usability of the resulting hypertext documents in terms of performance. One converter, for example, inserts navigation buttons by means of pointers to graphics files on the tool developer's own server, which imposes a performance penalty for the user trying to read the document.

Some Web-watchers and tool developers predict that in the future documents will be convertible to HTML on demand [DAGS 95]. The information provider need only maintain documents in their native electronic format — word processor, graphic, or whatever. The conversion tools will be powerful, efficient, and flexible enough to add the proper hypertext markup to documents automatically, without the need for manual tweaking. When a Web user requests retrieval of one of these documents, the server software will find the files, invoke the converter, and then send the newly-generated HTML output to the requesting client browser. Even this scenario requires document design decisions, however: which documents to convert ahead of time; which to make available for instant conversion; and how to set up the tool to generate the HTML version as desired.

A summary of the design decision steps needed for converting a linear document to hypertext is illustrated in Figure 21.

The first step is to define the decomposition of the document into pages. Each page will be a separate file on the server, and will be retrieved separately by the client's browser. The appropriate level of document component to use depends on its size, the number of non-textual elements included, and considerations of how the document can be read. It is not generally appropriate to use document pages, or other non-logical properties as a basis for decomposition, yet a balance must be found between consistency of file sizes and consistency of logical divisions. For example, in the conversion of this handbook to hypertext,

the primary decomposition unit is by table of contents entry (e.g., *3.3.1 Document Structure*). Yet the amount of text in each subsection ranges from a few paragraphs to several pages. The desire for a consistent interactive response won out over consistency in logical partitioning, so the longer subsections are split into multiple files, and the shorter subsections combined.

The second step of the conversion design process involves the non-textual portions of the document, such as figures and tables. Design questions include: choice of format; location with respect to the text; whether to include thumbnail (preview) images, or just captions linked to full-size graphics; specifying tabular data as preformatted, or capturing them as images. Non-textual design issues are presented in more detail in Section 3.4.

The next step is to identify intra-document links that will increase the users' understanding of the material, and their ability to navigate through the document. Appropriate uses of cross-links include any explicit references to other sections of the document, or to other documents on the server, as well as to acronyms, references, definitions, etc., as discussed in Section 3.3.1.

The last step is to design pages for the title and table of contents (TOC). The table of contents can be several pages, arranged hierarchically. Several tables of contents can be created at different levels, or to enable different reading sequences.

It is also worthwhile to review the text of the document to edit out phraseology that refers to the document as a physical, linear object. Standard paragraph-linking hooks and text flow indicators do not translate well to electronic media, especially hypertext, e.g., "as described above" "in the next section." Figure 22 illustrates another paper document remnant that a conversion tool processed into HTML.
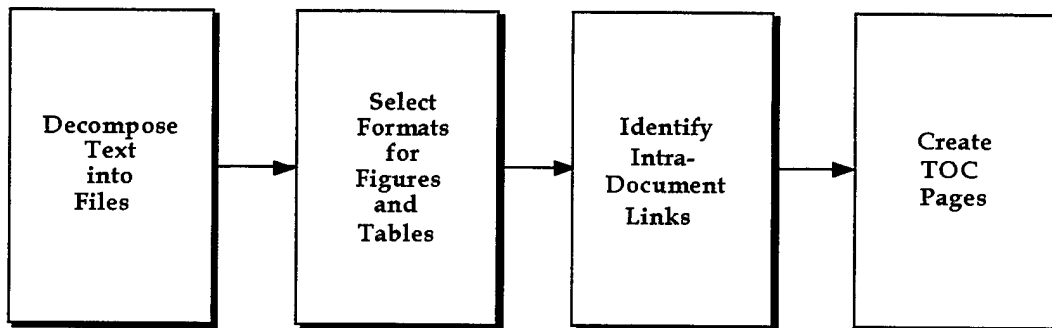
Figure 21: Design Steps for Converting Documents to Hypertext

File   Edit   View   Go   Bookmarks   Options   Directory   Help

Back   Forward   Home   Reload   Images   Open   Print   Find   Stop

Welcome   What's New   What's Cool   Questions   Net Search   Net Directory

Dr. Rudmik has developed several object–oriented repository technologies that have been used in compilation systems, software engineering environments, and programming systems.

Dr. Rudmik received B.A.Sc., M.A.Sc, and Ph.D. from the University of Toronto in 1970, 1972, 1976 respectively.

This page intentionally left blank.

References

[Atk92] Atkinson, Malcolm, et al., "The Object–Oriented Database System Manifesto." *Building An Object–Oriented Database System*, California: Morgan Kaufmann, 1992.

[Ban87] Banerjee, Jay, et al., "Semantics and Implementation of Schema Evolution in Object–Oriented Databases." *Proceedings of the ACM SIGMOD Conference*, (1987).

Document: Done.

Figure 22: "This page intentionally left blank."

The physical form of existing documents also affects the conversion process. If a document is only available as printed hardcopy, the most efficient method of creating an electronic version may be to scan the text, then process the scanned image with optical character recognition (OCR) software, rather than keying it in. The resulting text files will correspond to the original physical pages, not to logical divisions in the document.

An alternate strategy for publishing legacy documents on the WWW is to provide access via the Web to downloadable versions of the documents, rather than converting the entire document into browsable HTML files. The document can be stored in its native format, or saved as a Postscript file, which would be printable by more users. The information provider thus has more control over the appearance of the document. This approach is similar to the concept behind the Acrobat portable document format (PDF) [Adobe 94]. The difference is that PDF documents can be browsed electronically using the Acrobat reader software, instead of needing to be printed to be intelligible.

3.4 Multimedia Design

The ability to incorporate multimedia components into WWW information spaces adds interest and communication power to Web pages, and has been cited as one reason for the tremendous growth of the Web. Effective use of multimedia, however, requires careful tradeoff analyses, beginning early in the design process.

The term multimedia encompasses any type of non-textual material, such as images, audio, and movies. The addition of non-textual materials to Web documents affects design decisions because of the complexity they add. Unlike straight HTML-encoded text, which all Web browsers can interpret, multimedia uses a variety of data formats for information display

and retrieval, which differ among WWW browsers and platforms.

The performance tradeoffs for multimedia involve storage size, access times, and preview times. With a text-only HTML file, the browser starts filling in the screen while the page is being retrieved. This not only provides feedback that the information is being transferred and provides an indication of the retrieval speed, but also allows the user to start reading before the entire file is available. With most non-textual HTML files, however, browsers behave differently. For sound and video, the entire file must be present on the local machine before it can be processed or converted to the user-delivery medium, whether CRT or speaker.

Portability tradeoffs determine what types of multimedia can be included and still provide users with the desired information. Implementation tradeoffs for graphic images, such as the choice of viewers, data formats and compression techniques, are discussed in Section 4.1.3.

Inlined Images (Performance, Portability)
Graphical browsers access non-textual information by using HTML tags to point to a site identified by a URL. Many browsers allow inlined images to be selectively disabled so that the user sees only a generic image icon. This eliminates the time to download images and limits the delay in reading the text. Otherwise, at any time during browsing, when an inlined image is encountered, it is transferred to the local machine for display.

The technique of interlacing provides the ability to display an inlined image while it is still being retrieved, with progressively better resolution as the data file is downloaded. The user can get an idea of the image content, and can read any surrounding text, while the data is still being transferred. Interlacing alleviates most of the performance penalty that the inclusion of inlined images imposes on HTML pages, but only for users with browsers which can interpret the interlacing. Otherwise, even interlaced graphics must be completely retrieved before they are displayed to the user.

Another technique to increase responsiveness when retrieving documents that contain graphical images is to use thumbnail images. Thumbnails are reduced samples of the original image, usually 32x32 or 16x16 pixel representations of the original. By placing image data in a thumbnail format, a preview of the image can be conveyed, without incurring a severe response time penalty. If the user desires a better quality image, or more detail, the actual image can be retrieved by selecting the thumbnail which is linked to the full-size image. Image-laden documents such as catalogs, and browsers of information like satellite scans, typically have front ends which use thumbnail images.

Interactive Response (Performance)
The system response time when a user selects a link can vary from a few seconds to several minutes. Most browsers provide an activity indicator, such as the icon of the planet Earth rotating that Mosaic uses, to let the user know the request is being processed. In our experience, response times longer than three seconds start to become annoying. Waiting minutes for a link to be made and information to be retrieved becomes detrimental to usage.

Future browsers may provide more information about links, dynamically generated, to help users determine what the current system response time to complete a link will be before choosing to invoke it. Such information could include:
• Total file size of the requested information
• Current bandwidth capability
• Estimated time to complete transfer

Alternate Information (Portability)
Because some Web browsers, such as Lynx and CERN's WWW Line Mode Browser, do not have the ability to display images, document designers need to be aware of using images in such a way that they are critical to the understanding of the information in the document. The same awareness needs to be shown with other non-textual materials. Users who have non-graphical browsers will not be able to read or display the non-text files they encounter. For images, however, HTML

Data & Analysis Center for Software
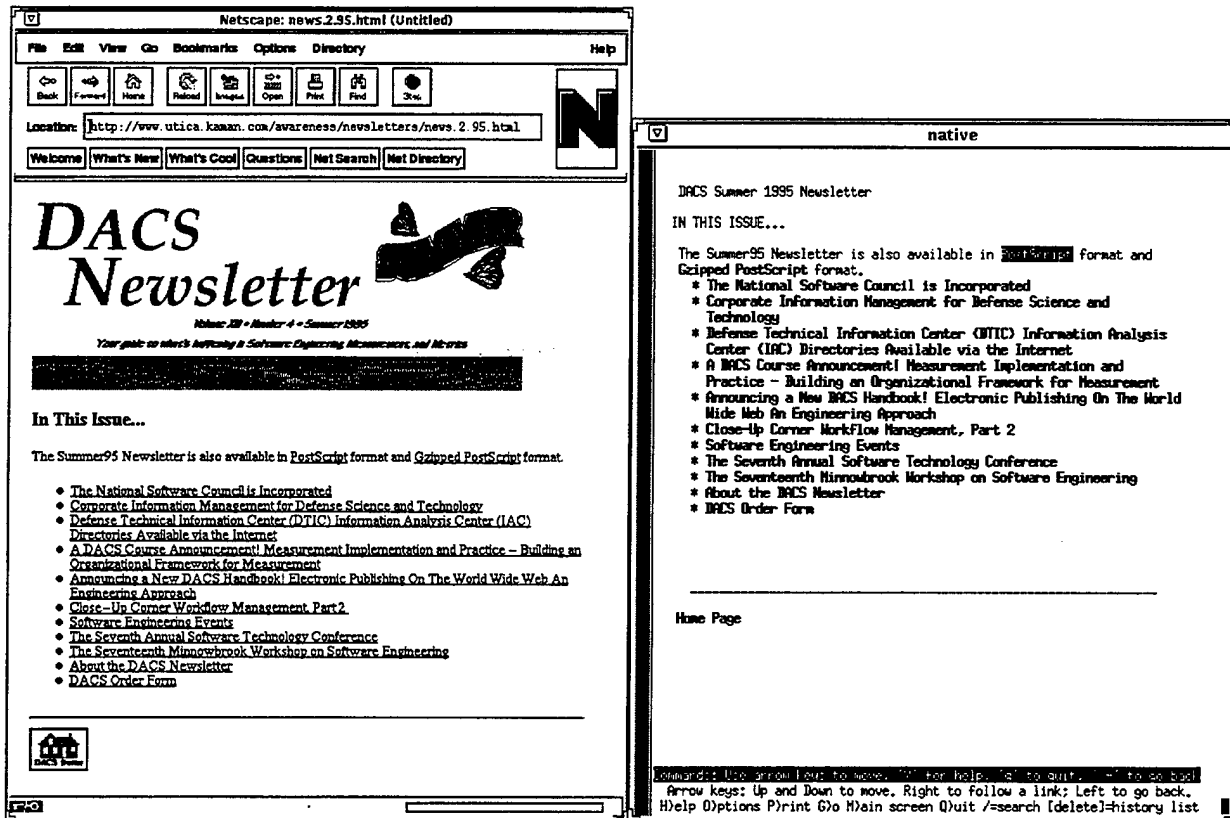
Figure 23: Image Display Comparison

provides an alternate construct, which can be used to display a textual description of the image instead of the image itself. Figure 23 illustrates the use of the alternate text, comparing a Netscape (Version 1.0N) display with the Lynx portrayal of the same file.

Another approach is to provide two versions of the files -- one with inlined graphics and one without. The tradeoff with this approach is increased overhead for creating, storing, and maintaining both versions. Maintenance efforts must be coordinated so that the two versions remain consistent with each other.

4 IMPLEMENTATION

Implementation refers to the actual creation of Web documents. After addressing the design issues of content and organization, determining the size and layout of individual pages, identifying the internal and external links, and deciding on the number and sizes of images to include, the developer will be ready to add HTML markup to documents, either manually or with the aid of tools. The intent of this section is to introduce techniques for using the markup language to create useful, readable and maintainable Web pages. Topics covered include HTML text markup syntax, the HTML forms interface, the inclusion of non-textual components in Web documents, and the incorporation of other information access interfaces into Web documents. Implementation process decisions regarding procedures, tools, services, and style are discussed in terms of their effects on the quality of Web documents. The importance of two other necessary steps in the HTML document production process, reviews and testing, is discussed as well.

Continuing the analogy between electronic publishing and software development, implementation is the step during which the program design is translated into software code, using a programming language. In the early days, this stage of software development was an art form. Much research has since gone into creating tools to help programmers translate designs into code more easily, and to create code that is more reliable, more testable, and more maintainable. Now, with powerful 4GLs and CASE tools, code can sometimes be generated automatically from the software design specifications. Many current HTML tool development efforts are seeking similar results.

The hypertext markup language and other implementation details in this section are presented from the perspective of what is needed for understanding HTML enough to be able to tweak and update documents. The assumption is that most Web information providers will not be writing HTML directly. After all, how many people still write assembly

code? As compilers became more capable, programmers moved beyond machine-level implementation details. The same is happening with HTML. Since the first draft of this handbook was prepared, tools have become available that allow people to create Web pages without knowing any HTML syntax.

This does not mean it is no longer necessary for anyone to learn about HTML. Most documents will need to undergo a few iterations of markup and testing before the desired result is achieved. Implementors interested in creating and maintaining high quality pages that meet specific publishing objectives need sufficient knowledge of HTML markup to be able to refine, update, and customize the markup applied by automated tools.

4.1 HTML Implementation

The information presented in this section describes the implementation of three types of Web page components: text, forms, and multimedia, then mentions some of the newly emerging applications such as interactive pages, database interfaces, imagemaps, and three-dimensional markup. Technical details can be found in the references in the Bibliography, in Appendix B.

4.1.1 Text Markup Summary

Hypertext Markup Language (HTML) is the name of the encoding scheme used to format information for publication on the World Wide Web. Web browsers read and display hypertext documents that are coded in the hypertext markup language. HTML is compatible with the Standard Generalized Markup Language (SGML), which is the standard for formatting electronic text among desktop publishing applications. Because Web documents must be compatible with many platforms and browsers, however, HTML provides only a limited number of SGML formatting constructs.

HTML is used to specify the logical organization of a document, including any hypertext links. The process of marking up an HTML file consists of adding "tags" to the content that communicate information about its structure. HTML includes tags to specify basic formatting commands as well. The tags in an HTML file are interpreted by the WWW client browser, which follows link references to retrieve other HTML files or resources as needed, and interprets the formatting tags to determine how to display the files on the user's screen.

There are many good, detailed reference guides to HTML syntax, and they are proliferating. Even commercial presses are publishing HTML books and programming manuals now. Some of the most widely-referenced on-line guides are shown in Table 7. Others are listed in the Bibliography (Appendix B). In addition to these HTML references, Appendix E contains a summary list of tags, and a sample file that contains the elements in HTML Level 2, structured into a "test pattern" that can be used to investigate the behavior of any browser and platform combination. Appendix E also contains a formatted version of how the test

pattern appears on one browser and platform combination (Mosaic 2.5 Beta 4 for the X Window System).

In HTML files, the tags are set off by angle brackets <...>. Most of the formatting tags occur in pairs, to mark the beginning and end of a text block. For example the title of a document would be coded as: <TITLE>Internet Tools</TITLE>. Paragraphs of HTML text are separated by the single tag <p>. Other basic tags indicate various levels of headings: <H1>..</H1>, <H2>..</H2>, <H3>..</H3>, etc.; emphasis: <STRONG>.. </STRONG>; bulleted lists: <UL><LI>..<LI>..<LI></UL>; and so forth.

HTML tags that define hypertext links are called anchors, and are identified by <A>...</A>. The text between the tags is displayed on the screen as a link (links may be underlined, or highlighted, or numbered, depending on the browser and platform). The URL of the link destination is coded in the first half of the anchor tag, in quotes, thus <A HREF="http://www. utica.kaman.com/">DACS Home</A> would be used to specify a link to the top-level DACS home page.

Table 7: HTML References

TITLE	URL (current as of August 1995)
A Beginner's Guide to HTML	http://www.ncsa.uiuc.edu/General/Internet/WWW/ HTMLPrimer.html [NCSA 94a]
An Introduction to HTML	http://www.vuw.ac.nz/non-local/gnat/ www-html.html [Torkington 94]
HTML Documentation	http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/ index.html [Graham 94]
HTML Tutorial	http://fire.clarkson.edu/doc/html/htut.html [Horn 94]
How to write HTML files	http://kcgl1.eng.ohio-state.edu/www/doc/htmldoc.html [Flynn 94]
Style Guide for On-line Hypertext	http://www.w3.org/hypertext/www/Providers/Style/ Overview.html [Berners-Lee 94c]
Composing Good HTML	http://www.cs.cmu.edu/~tilt/cgh [Tilton 93]
HTML Reference Guide	http://www.w3.org/hypertext/WWW/MarkUp [Connolly 94]
HTML Quick Reference	http://www.cc.ukans.edu/Lynx.Help/HTML.Quick.html [Grobe 94]

Data & Analysis Center for Software

```
<TITLE>Title</TITLE>

<IMG ALIGN=middle SRC="../images/wrench.gif"><p>

<UL>
<LI><A HREF="MGRM/MGRM.html">Metrics Guided Risk Managemen (MGRM)
Tutorial</A>
<LI><A HREF="oasys/oasys.html">OASYS</A> -Workflow Automation System
<LI><A HREF="aces.html">ACES</A> - Ada Compiler Evaluation Suite
<LI><A HREF="camp.html">CAMP</A> - Common Ada Missle Packages
<LI><A HREF="firststep.html">FirstSTEP - Metrics Analysis</A><p>
<A HREF="ftp://ftp.utica.kaman.com/pub/FirstSTEP/FirstStep.tar.Z"><IMG
ALIGN=middle SRC="../images/modem.gif"></A>Click here to download the
FirstSTEP tar file!  <A HREF="tools.probs.html">Having Problems?</A><p>
<LI><A HREF="afatl.html">AFATL</A> Compiler - Air Force Armament Laboratory
(AFATL) ADA Compiler and ADA Interactive Debugger
<LI><A HREF="goel.html">Goel-Okumoto</A> - Goel-Okumoto Software Reliability
Model
<!-- <LI><A HREF="iemcap.html">IEMCAP</A> - Intersystem Electromagnetic
Compatability Analysis Program-->
<LI><A HREF="acs.html">ACS</A> - ADA Compilation System
</UL><p>

<HR NOSHADE>
<A HREF="../index.html"><IMG ALIGN=bottom SRC="../images/home.gif"
ALT="Back to the DACS Home Page"></A>
```

Figure 24: A Web Page and its HTML Source File

The NAME tag allows the destination of a link to be more finely specified, so that the browser displays a particular part of a file, rather than always starting at the first line. This is useful for creating internal links within a file, as well as links to specific points of another file. The syntax for the destination anchor is <A NAME="destination">Text</A>. The format for the source anchor is the same, with the destination added, e.g., <A HREF="file #destination">Text</A>. If the link is to elsewhere in the same file, the filename can be omitted, e.g., <A HREF="#destination">Text </A>.

The HTML syntax for links to images includes the URL of the image file. Only a single tag is needed. Thus <IMG SRC="http://www.utica .kaman.com/awareness/newsletters/images/ home.gif"> indicates a graphics interchange format (GIF) file in the DACS Newsletters images directory.

Figure 24 shows an example of a page of information from the WWW that includes both images and hyperlinks. The current version of this page can be reached from the DACS home page. The left side of the figure shows how the page looks using the Netscape browser. The Location window on the screen display shows that the name of the file from which the browser produced this page is "tools.html." That file, marked up with HTML tags, is shown on the right side of Figure 24. Most browsers provide a way to see the HTML markup for any Web page. Using Netscape, it can be seen by choosing the "Source" option on the "View" pull-down menu. With Mosaic, the "View Source" option is on the "File" pull-down menu.

Notice the title tag at the beginning of the file, the inlined image specification for the wrench picture, and the bulleted list of items, each of which also contains a link to another document.
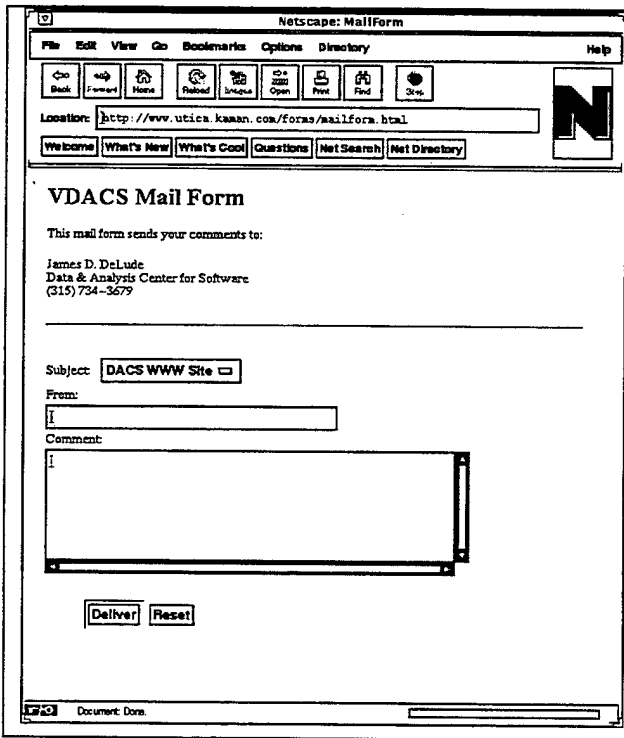
HTML is an evolving language. All browsers can interpret the lowest level (minimum set) of HTML tags. Most graphical browsers can interpret the features at Levels 1 and 2. The exact definition of what will be incorporated in

Table 8: HTML Levels

LEVEL	CHARACTERISTICS	SAMPLE BROWSER
HTML Level 0	Minimum conformance level Mandatory. Headings, lists, anchors, etc. Least presentation differences between platforms	CERN's WWW Line Mode Browser
HTML Level 1	Phrase-level markups and image implementation Images Emphasis	Chimera
HTML Level 2	Forms support Requires greater implementation effort	Mosaic
HTML Level 3 proposed elements	Figures and images Generic Highlighting Tag Tables Mathematical equations Nested divisions and containers HTML/TEI conversion support	(under development)
Browser- Specific Extensions	Text-wrapping around graphics Blink highlight	Netscape

the next version of HTML, known as HTML3, is still being developed. Table 8 summarizes the different levels of HTML. More information about the standardization process and HTML3 can be found in Section 7.4. In addition to the official standardization efforts, the capabilities of popular browsers, such as Mosaic until recently and Netscape now, become defacto standards. Although the use of such features, known as "browser-dependent HTML extensions," reduces the portability of pages, there are many situations where the control or


```
<TITLE>MailForm</TITLE>
<H2>VDACS Mail Form</H2>
This mail form sends your comments to:<p>
<DL>
<DT>James D. DeLude
<DT>Data & Analysis Center for Software
<DT>(315) 734-3679
</DL><p>

<HR>
<Form Method=POST Action=/cgi-bin/form-mail.pl>
Subject:<select name=subj><option selected> DACS WWW Site</select>
<BR>
From:<BR>
<INPUT Name=username Size=42><BR>
Comment<BR>
<TEXTAREA NAME=comments ROWS=8 COLS=60></TEXTAREA><p>
<B>
<UL>
<Input Type=submit Value=Deliver>
<Input Type=reset Value=Reset> <BR>
</UL></B>
</Form>
```

Figure 25: A User Input Form and its HTML Source File

Data & Analysis Center for Software

capability added outweighs any browser-specific limitations. Many pages on the Web are labeled "Netscape-enhanced" to indicate incorporation of browser-dependent features.

4.1.2 HTML for Forms

Using the HTML forms interface, Web authors can create HTML documents that contain forms to be filled out by users. The HTML forms interface incorporates text input fields that allow interaction between the user and the server through the Common Gateway Interface (CGI). The CGI is the standard by which external programs (often called gateways) interface with hypertext transfer protocol (HTTP) servers. CGI programs act as gateways between the HTTP server and databases, or between the server and local programs or document generators. When a user fills out the form and "presses a button" indicating the form is to be submitted, the information on the form is sent to a server for processing by a CGI program. The server will usually prepare an HTML document using the information supplied by the user and return it to the client for display. An example of a simple form and its HTML file are shown in Figure 25. The left side of the figure is the Netscape-interpreted display of the HTML markup on the right.

CGI programs can be designed to handle many interactive applications. The services provided can include anything from querying databases to populating them. The CGI program can accept an information request to a database, create the proper database query, retrieve the results from the database management system, assemble them into an HTML document, and return this document to the browser. The forms interface is implemented with the following tags:

- <FORM> . . . </FORM> Defines a form. The action attribute specifies the URL location of the program that will process the form
- <INPUT> Defines an input field (no ending tag). Several types are defined
- <SELECT> . . . </SELECT> Defines a select field. Requires an <OPTION> element for each item in the list
- <OPTION> Defines the possible values for a field option within the <SELECT> element
- <TEXTAREA> . . . </TEXTAREA> Defines a text area where the user may enter text data

On the server side, forms are processed by scripts which take what the user has typed in the form fields as input variables. The scripts can be written in any number of programming or procedural languages, including C/C++ programs, Bourne Shell Scripts, C Shell Scripts, Tool Command Language (TCL), Practical Extraction and Report Language (PERL), or other executable programs. On the server machine, the script programs are located in a directory named "cgi-bin." This tells the browser program that they are to be executed rather than interpreted for display. A sample of the script for the mail form shown in Figure 25 is reproduced in Figure 26.

```perl
#!/usr/local/bin/perl

# This should match the mail program on your system.
$mailprog = '/usr/lib/sendmail';

# This should be set to the username or alias that runs your
WWW server.
$recipient = 'vdacs-admin@utica.kaman.com';

# Print out a content-type for HTTP/1.0 compatibility
print "Content-type: text/html\n\n";

# Print a title and initial heading
print "<Head><Title>Thank you</Title></Head>";
print "<Body><H1>Thank you</H1>";

# Get the input
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});

# Split the name-value pairs
# print $buffer;
@pairs = split(/&/, $buffer);

foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);

    # Un-Webify plus signs and %-encoding
    $value =~ tr/+/ /;
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
```

```perl
    # Stop people from using subshells to execute commands
    # Not a big deal when using sendmail, but very important when using
UCB mail (aka mailx).
    # $value =~ s/~!/ ~!/g;

    $FORM{$name} = $value;
}
# Now send mail to $recipient

open (MAIL, "|$mailprog $recipient") || die "Can't open
$mailprog!\n";
print MAIL "Reply-to: $FORM{'username'} \n";
print MAIL "Subject: WWW comments (Forms submission)\n\n";
print MAIL "$FORM{'username'}, sent the following\n";
print MAIL "Comment or question about The DACS's WWW
server:\n\n";
print MAIL "------------------------------------------------\n";
print MAIL "$FORM{'comments'}";
print MAIL "\n------------------------------------------------\n";
print MAIL "Server protocol: $ENV{'SERVER_PROTOCOL'}\n";
print MAIL "Remote host: $ENV{'REMOTE_HOST'}\n";
print MAIL "Remote IP address: $ENV{'REMOTE_ADDR'}\n";
close (MAIL);

# Make the person feel good for writing to us
print "Thank you for sending comments or questions to <I>The
DACS</I>!<P>";
print "<A HREF=\"/index.html\"><IMG ALIGN=bottom
SRC=\"http://www.utica.kaman.com/images/home.gif\"></A>";
```

Figure 26: A Script for HTML Form Processing

4.1.3 Implementing Multimedia

Although many factors related to the display of non-textual materials are browser-dependent, the same implementation process decisions are required. Incorporation of multimedia requires the Web author to choose from among different format, compression and quantization techniques, for best compatibility with users' browsers, platforms, access to special viewer software, and communications bandwidths. As emphasized in the design chapter, the primary tradeoffs are portability and performance versus the size, quality and detail of multimedia components.

Assuming the goal is to provide users with as much information as possible in non-text files, tradeoff analyses are necessary to determine how much of the original information may be lost while still conveying the desired message. The information provider must take into account the type of message, its data representation and the available bandwidth.

The first step, therefore, is to evaluate the information for possible reduction. Images and audio files usually can be reduced 50% or more with little noticeable degradation. The number of colors in an image can be reduced to the minimum needed to convey the information.

Other multimedia implementation techniques include:

- Providing alternate text to replace images, to accommodate text-only browser tools
- Providing viewer applications for the site, to avoid wasting network bandwidth and the user's time transferring images and audio that can not be interpreted
- Providing a link to the "viewers" section of the NCSA documentation that describes the viewers needed, if any
- Using thumbnail images to allow users to preview images before retrieving them

Formats
There are many different image formats to choose from. NCSA Mosaic images are restricted to X Windows Bit Map (XBM) and

Graphics Interchange Format (GIF) files, however. Since XBM is not usually compressed or supported on all platforms, GIF images will be more portable.

For speech narration, a sampling rate of 8 kilohertz (kHz) as normally provided on Sun workstations, is adequate. Sampling rates of from 5.6 kHz to 48 kHz are available on the Macintosh and machines running Microsoft Windows, but 8 kHz will provide maximum compatibility. When using higher sample rates, the implementor should consider minimizing the file size or providing the user with references to compatible external viewing tools.

The only video format currently supported on all platforms is the Motion Photography Experts Group (MPEG) standard, so use of that format will provide maximum portability among platforms.

Viewers
Viewers are programs which can be automatically launched by the WWW browser.

When a link is selected that contains non-HTML data, such as an image file, the file is transferred from the server machine to the client's machine. Once the file transfer is complete, another program is launched to process the image. Viewer programs are available to handle all the types listed in Table 9.

Table 10 lists the suffixes which are recognized by some browsers as non-text files. Upon encountering these file types an external viewer application will be launched when the file transfer is complete.

The browser will launch an appropriate viewer based on the suffix of the file. These relationships can be defined by users in their browser preferences setups. Viewers are kept external for many reasons, including size, proprietary software, performance, flexibility and adaptability. If a user never needs to listen to audio files then there is no reason to consume browser code, disk space and memory use for this feature.

Table 9: Common Electronic Representation Data Types

Type	Electronic method of translation or conveyance
Images	Information by pixel elements
Text	ASCII character codes
PostScript	PostScript instructions, embedded images and text
Audio	Digitized sound samples
Multimedia	Sequences of image frames and time-correlated audio

Table 10: Common Viewers and Platform Compatibility

Extension	Macintosh	Windows	Sun UNIX X-Window
.aiff	Built in	N/A	N/A
.au	SoundMachine	Wham	Sun AudioTool
.gif	NCSA Mosaic	NCSA Mosaic	NCSA Mosaic
.jpg	JPEGVIEW	Lview	XView
.mpg	Sparkle	Mpegplay	mpegplay
.mov	MoviePlayer	MediaPlayer	N/A
.ps	GhostScript	GhostScript	GhostScript/pageview
.tiff	JPEGVIEW	N/A	XView
.xbm	N/A	N/A	Built in X-Windows
.Z	MacCompress	gzip (decomp)	decompress
.gz	gzip	gzip	gzip

Compression

Compression is used to reduce the physical size of files. Having smaller files allows for a more responsive interactive session with the user. Two types of compression may be employed, lossless and lossy. Lossy compression will provide compression ratios of 10:1 to 1000:1, but removes some of the original source information. For images and sound, it is typical to use lossy compression. The user can interpret the information without any loss of meaning. Typical lossy compression techniques are employed with the Joint Photographers Expert Group (JPEG) standard for still images, the MPEG standard for video, and most audio sampling techniques. Lossless compression is still the standard for textual information, whether in raw form or formatted, e.g., using PostScript or LaTeX.

PostScript files are usually compressed with UNIX compress or gzip, although gzip may not be available everywhere. Often both compression formats are provided: a UNIX .Z file and a gzip .gz file.

Each of the compression techniques in Table 11 is the result of an optimization for a specific data type.

Table 11: Compression Techniques

FORMAT	LOSSY	DESCRIPTION
AIFF	YES	AUDIO, Similar to AU with Apple Macintosh specific orientation.
AU	YES, minimal	AUDIO, Digital audio data represents a quantized approximation of an analog audio signal waveform. In the simplest case, these quantized numbers represent the amplitude of the input waveform at particular sampling intervals. In order to achieve the best approximation of an input signal, the highest possible sampling frequency and precision needs to be used. However, increased accuracy comes at a cost of increased data storage requirements. AU files provide audio quality comparable with analog telephone service, with Sun-specific orientation.
GIF	NO	IMAGE, 8 bit per pixel encoding, this allows 256 colors and provides typical compression of the raw image by 2:1 to 10:1.
GZIP	NO	GNU Project's compression program. It can also decompress UNIX Compress files (.Z).
HTML	NO	HyperText Markup Language derived from a Standardized Generalized Markup Language (SGML) Document Type Definition (DTD).
JPEG, JPG	YES	IMAGE, 24 bits per pixel encoding, this allows images with 16.7 Million colors to be compressed. Typical ratios here depend upon the amount of information or quality the user needs. Normally with a 75% quality value compression will be on the order of 10:1 to 100:1.
MOV	YES	Quicktime movie. A standard created by Apple Computer for display and manipulation of digital video data. Quicktime incorporates image compression and audio compression of various types.
MPEG, MPG	YES	Motion Photography Experts Group standard for video, incorporating unique compression algorithms.
PS	NO	PostScript. The PostScript print file format is a programming language with powerful graphics primitives for describing printed pages. It has the ability to store embedded images. This is one the most common file formats now being used for storing documentation. It allows any PostScript compatible printer or PostScript viewer application to display and print the information.

TIFF	YES/NO	Tagged Image File Format. Specifically developed as a standard for interchange of digital image data. TIFF itself describes the format of the TIFF file. Under the TIFF specification any compression or data storage techniques may be employed.
TXT	NO	ASCII text.
XBM	NO	UNIX X-Windows standard for image files. It supports black and white, and color images with no compression.
Z	NO	UNIX Compress. This is a standard for compressing files and is normally the way files are stored in archives for things such as PostScript.

Communications Bandwidth

A user browsing the Web may encounter hundreds of links to non-textual files for images, sound and movies. Ideally these encounters would be navigated instantaneously, giving results similar to switching channels on a television. In reality, the WWW system is more like a telephone system, where only a limited number of phone calls may be made through the system, due to limitations in bandwidth and channels. Currently browsers rely on traditional TCP/IP, SLIP/PPP communication protocols to move information between nodes on remote and local computers. References to more information on protocols can be found in Appendix B. This brief overview of communication transports is provided to help define what the limits are, and to quantify the effect of various implementation tradeoffs on performance. The communication interfaces Table 12 are specified in bytes per second for ease of comparison. A byte uses from 8 to 11 bits to transmit, depending on the protocol overhead.

Each of the connection techniques in the Table 12 is also influenced by the amount of traffic being handled on the Internet. Even with a high speed link of 56 KBS, if 1000 users are transferring files through the same routes, the effective speed may only be 1,500 bytes per second. Figure 27 compares transmission times of various file sizes for a few of these communication methods.

Quantization Example

The following example is provided to illustrate the appearance vs. performance tradeoffs required in implementing image data files on the WWW, and to stress that all formats and techniques available for image compression and quantization have side effects. A series of operations on a sample image are illustrated in Figure 28. (Note that the production processes required to generate and print the paper version of this handbook have further degraded

Table 12: Communication Interfaces

LOCAL	Locally connected machines via ethernet typically communicate at 100-400,000 bytes per second.
T1	A type of telephone line, that provides site to site service with an effective bandwidth of approximately 150,000 bytes per second. University and large commercial organizations typically have this service.
ISDN	6,400 bytes per second service. Integrated Services Digital Network (ISDN) has both voice and data at 6,400 bytes which can be combined to 12,800 bytes per second data.
56 KBS	5,600 bytes per second. These are usually leased lines. Smaller organizations (100-1000 people) usually have this service.
MODEMS	30-2,800 bytes per second. Modems are used to make permanent and temporary connections to service providers.

Data & Analysis Center for Software

53

TRANSFER (WAIT) TIME COMPARISON
All times in seconds

METHOD	RATE (bytes/sec)	FILE SIZE			
		1,000(1)	10,000(2)	100,000(3)	1,000,000(4)
Local or TCP/IP	200,000	<1	<1	<1	5
56 KBS	5,600	<1	2	20	180
14.4 KBS	1,400	<1	8	72	715

(1) Typical WWW Home Page.
(2) Large WWW Home Page.
(3) Typical image file, PostScript file, audio file.
(4) Typical movie file, large PostScript or audio file.

Figure 27: Transfer Times
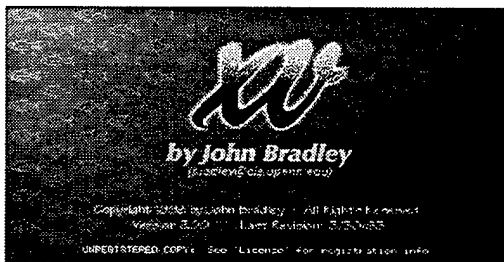
Image 1
IMAGE GEOMETRY: 482x274
BITS PER PIXEL: 8
RAW FILE SIZE: 33,791 bytes
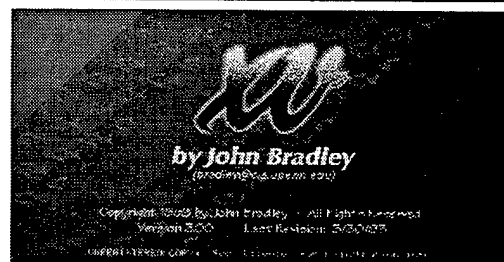
Image 2
IMAGE GEOMETRY: 482x274
BITS PER PIXEL: 4
RAW FILE SIZE: 19,523 bytes
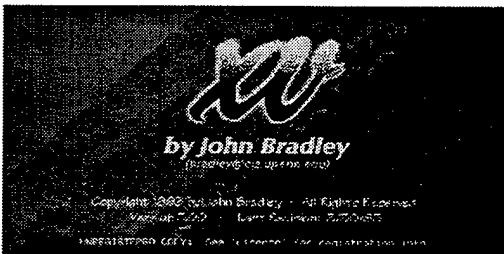
Image 3
IMAGE GEOMETRY: 482x274
BITS PER PIXEL: 3
RAW FILE SIZE: 15,261 bytes

Image 4
IMAGE GEOMETRY: 482x274
BITS PER PIXEL: 1
RAW FILE SIZE: 4,479 bytes

Figure 28: Image Quantization Example

Data & Analysis Center for Software

Table 13: Summary of Image Operations

IMAGE	SIZE	COLORS	QUALITY	% of Original
Original	132,068	256	100%	100.0%
Image 1	33,791	256	100%	25.9%
Image 2	19,253	16	color loss	14.6%
Image 3	15,261	8	color loss	11.6%
Image 4	4,479	2	color/image loss	3.4%

the images into black and white approximations of the examples in the electronic version.) The statistics for the original image (not shown) were:

IMAGE GEOMETRY: 482x274
BITS PER PIXEL: 8
RAW FILE SIZE: 132,068 bytes (482x274x8 / (8 bits per byte))

Image 1 is the original image in GIF format, with no information loss. With 8 bits per pixel, 256 colors are allowed. Image 2 is quantized for reduced colors. This image has been reduced to 16 colors or 4 bits per pixel, still in GIF format, but some color information loss has occurred. Image 3, quantized still further to reduce the number of colors, has been reduced to 8 colors, or 3 bits per pixel. It is still in GIF format, but most of the color information has been removed. Image 4 has been quantized to the minimum number of colors. At 1 bit per pixel, only two colors are left. Although still in GIF format, the color information has been removed.

Table 13 summarizes the differences in the images. The tradeoff is between image quality and system responsiveness, due to the size of the image file. The implementor needs to choose the minimum size image that will still convey the necessary information.

4.1.4 Other Interfaces to HTML

One of the greatest areas of change since the initial draft of this handbook was written is in the new types of data and applications that are becoming available on the WWW. In the past six months, many tools have been developed, providing interfaces that move away from just

text and graphics toward more interactive applications. A comparison of the topics on the WWW FAQ pages [Boutell 95] between October 1994 and July 1995, reveals the following new types of Web interfaces and multimedia tools:

- Virtual Reality Markup Language (VRML): for creating three-dimensional Web pages
- Java: Sun's language for creating applications that can be transported across the Web
- Imagemaps: for embedding links within areas of an image
- Transparent images: changing the color of an image to match the user's background
- Interlacing: allows gradual display of inlined images
- Text-Wrapping: fills in the space around an image with paragraph text

Tools that interface with databases across the Web, combined with Wide Area Information Server (WAIS) capabilities are greatly increasing the type and amount of information that users can retrieve from individual sites and from the WWW as a whole. With the database systems, what the server returns in response to a user's request is not a pre-existing HTML file, but a newly generated file, that is constructed from the results of a database query. This type of interface, combined with HTML forms, allows for sophisticated interaction between WWW users and information spaces.

4.2 Implementation Process

The difference between creating an original page of Web information and converting an existing document into a set of interconnected Web pages is less important in the implementation phase than in design. There are, however, implementation choices, such as

the selection of tools, which have quality tradeoffs.

4.2.1 Document Production Options

Converting an existing sequential document to HTML may be conceptually more difficult than creating one specifically for implementation in HTML, because HTML is best used to define the logical structure of a document rather than its physical appearance, and therefore the converted document will not retain the look of the original.

It is possible to produce HTML files with a standard text editor or word processing program. The process is tedious, however, and the probability of making errors is high. The task of creating or converting documents to HTML is made easier with an HTML editor, regardless of development platform or development method. The editor removes the burden of having to memorize the syntax of HTML, or type in the tags precisely. An editor or word processor with HTML tags embedded in it would be useful for creating HTML files as a document is being written.

Currently, the most common method for converting an existing document to HTML is to use a conversion tool. For an existing document, an HTML conversion tool that understands and retains the logical and physical structure information embedded in the existing document format, would be more efficient. But most conversion tools can not create a complete HTML document from an existing word processor document, automatically breaking it into multiple files and creating links to each component file and each included image. Because a perfect conversion tool doesn't exist, conversion requires an additional step where a text editor (or HTML editor) is used to finish converting the individual HTML files into a complete HTML document. The newer authoring environments do a better job of shielding implementors from HTML, but for large documents the implementor must still make decisions about how best to break them apart and link them together.

Regardless of how the HTML document is created the first time, it will likely need refinement and updating during its publication life. Often the most efficient way to update a document is with an HTML editor, especially if the changes are minor in proportion to the size and complexity of the entire document or file.

Sequential Versions of Hypertext Documents
To increase the usability of electronic publications, it may be desirable to provide a sequential version of the text in addition to the hypertext version. For documents that begin as sequential text and not hypertext, the prospect of maintaining two versions will not impose a big additional burden. The original, sequential version can simply be made available via FTP or other Internet transfer tool.

The hypertext versions of complex or large documents, such as technical reports or newsletters, that are implemented as collections of linked and nested files, are candidates for multiple versions. A user may wish to download or print out the text of an entire document for future reference, or further distribution. But retrieving and ordering all of the text in a hypertext document with several levels of links will be very time-consuming, if not nearly impossible. If, however, a sequential version of the document is available, the user has the option of retrieving and printing the complete document in a readable format. The same effect can be achieved by providing a PostScript version along with the hypertext, but that assumes that the end-user has access to a PostScript compatible printer.

Considering the attributes of a document will help determine whether or not providing a separate sequential version is worthwhile. An inherently non-linear document, such as a home page or a directory of external resources, probably does not need a separate downloadable version. Knowledge of how users will be reading the documents will help in making the proper tradeoff between the value of providing alternate forms of documents and the costs of storing and maintaining multiple versions of the same information on the Web server.

Data & Analysis Center for Software

4.2.2 HTML Tools

Development of new and better tools to support HTML is continuing at a rapid pace. HTML tools can be classified into several groups:

- Converters that add HTML markup tags to existing files in various formats
- Editors that produce HTML markup tags as a file is being created or revised
- Authoring environments that support the entire process
- Administrative-type tools useful for testing, maintaining and monitoring HTML files
- Programmer-style utilities for managing various aspects of HTML implementations

Document conversion tools vary from platform to platform and from format to format. A typical conversion program will provide some but not all of the necessary HTML tags. Some conversion programs work to convert generic file formats such as Rich Text Format (RTF). Others are more specific, such as mifmucker, which converts FrameMaker MIF files. Several templates are available for standard word processing programs which convert the word processor formats to HTML files. With a converter, the process is usually a cycle consisting of three steps, which continues until the document is deemed acceptable by its creator:

- Convert the document
- View the document
- Edit the document

A listing of tools is maintained at the *W3 and HTMLTOOLS* page on the Web [Berners-Lee 94b]. Many of the tools listed are available "free" off of the Internet. The tools vary considerably in their level of power and sophistication. Some are full-fledged software products; others are simple UNIX sed or awk scripts that someone found useful enough to share with the world. Other Internet sources of information on tools can be found in the sources listed in Appendix B.

The usefulness of any particular conversion tool or script will depend on the format, size and complexity of the document being converted. Appendix C of this handbook presents assessments of some conversion tools, based on experience with them at the DACS.

A class of potentially transformative tools, now appearing, is represented by Acrobat [Adobe 94]. Acrobat is a joint effort between Adobe, makers of PostScript, and Apple. The idea behind these tools is to develop a single front-end processing tool that integrates all the document types, and allows automatic translations among them. With such a tool, the conversion to HTML (or whatever the current WWW publishing standard is) would be a single-button operation. Therefore, publishers need only create and maintain their electronic documents in their native formats, because conversion can be accomplished automatically.

4.2.3 HTML Services

An emerging Web phenomenon is authoring and conversion services. There is a wide range of these, from simple file conversion to full-fledged consulting and Web site operation businesses. Information providers at several sites have offers to create Web pages for users. One set-up asks users to send files for conversion via E-mail to a specified address. The first line of the mail message is a command to execute one of the site's available conversion programs, e.g., "execute rtftohtml," and the rest of the message is the file in whatever format that conversion program works from, in this case, Rich Text Format. The provider, receiving the E-mail request, converts the incoming file, and mails the HTML file(s) back to the sender. Other schemes use FTP instead of or in addition to E-mail, especially for transfers involving non-text files.

Another site has pages similar to the HTML test pattern included in Appendix E, that includes multimedia components that can be used by WWW site administrators to test their software installation, and for WWW users to get demonstrations of Web capabilities.

4.2.4 HTML Style

Concern with issues of style when authoring World Wide Web documents reflects the next step in the evolution of electronic publishing. Style refers to making choices among different ways of achieving the same result, for reasons other than merely what works. If the first step was concerned with just getting the HTML right, then the next step is concerned with improving quality, by paying attention to how the markup is applied. This is analogous to the definitions of proper programming styles in the early 1970's, that included both readability concerns such as indenting, comments and header blocks, and maintainability concerns, such as using standard algorithmic structures.

The amount of literature describing the use of HTML is increasing, in both on-line and hardcopy sources. Many of the introductory HTML guides include sections on style. Unlike the issues raised with document design, and the open issues presented in Chapter 7, the advice and suggestions for good HTML style seem to be converging to a coherent set of best practices. The collection presented in this section, drawn from the HTML literature and from the DACS' experience with both electronic publishing and with software quality issues, is categorized in terms of maintainability, portability, and usability.

In all cases, markup style must be tailored for the particular situation. If a document is a short-lived "news flash," for example, a developer may not care that it is hard to maintain, because it will be gone before it needs updating. In another case, a document may have no requirements for portability among browsers. One global style suggestion is to be consistent, because consistency increases usability, maintainability, and portability.

Like software code, HTML files are written once but read many times, so efforts expended to make it readable, by both users and maintainers, are worthwhile. Style guidelines for increasing the readability and usability of electronic documents include:

- Sign and date each separately retrievable page. Then users know how old the information is, and can get further information, if desired, by contacting the author.
- Indicate the status of the document. This will provide the user with more clues about the quality, validity, stability, etc. of the document.
- Make titles, etc. independent of their surroundings (context-free). This relieves users from having to follow a prescribed reading sequence in order to understand a document.
- Avoid describing implementation details in the text. This shields the user who just wants information from having to know or understand jargon.
- Incorporate navigation aids. This helps users find what they need more easily, and keeps them from feeling lost.
- Select filenames carefully. They can be used to add information, which can aid navigation.

Style guidelines for increasing the maintainability of HTML files include:
- Use relative pathnames within a document, and absolute pathnames for links to other documents, assuming the document is contained in a single directory. Then parts that are likely to move together can be kept in synch without a lot of editing.
- Allow and encourage feedback. Users can help focus maintenance efforts by pointing out problems, or indicating areas of most interest.
- Include comments using the HTML comment tag, <!--comment-->, for information that need not be displayed to the user. Just like in code, they help the maintainer understand how and why the markup was implemented. But unlike software source code, the comments are available to any browsing user who elects to view the HTML source.

Style guidelines for increasing portability among browsers and platforms include:

Data & Analysis Center for Software

- Use logical tags rather than specifying physical formats, e.g., <EM>emphasis</EM> instead of <I>italics</I>. This allows each user's browser to interpret the HTML markup file in the best way for the user's platform.
- Strive for device independence. For example, avoid using "click here" which assumes the user has both a mouse and a graphical interface.
- Use only official HTML constructs, avoiding both those that are not supported and those that have become obsolete, to ensure that HTML files will be displayable by any legitimate browser.

The reference *Composing Good HTML* [Tilton 93] contains specific examples of portable and non-portable HTML constructs. Tilton emphasizes, for instance, that the <p> tag does not mark the end of a paragraph, but is used to separate blocks of text. For continued portability, however, note that HTML3 proposes transitioning to the use of paired paragraph tags, <p> and </p>, for better compliance with SGML. The portability of some constructs may not be apparent until a page is tested with different platforms and browsers. For example, when heading tags are used in conjunction with list tags, the relative order of the <H#> and <LI> affects how they are displayed using Mosaic on a PC, but does not matter using Netscape.

The more sophisticated HTML authoring and conversion tools that are becoming available have their own definitions of proper HTML style embedded in them. After using one of these tools to create an HTML document, it may be necessary to undo some of what the converter did automatically, replacing the converter's choice of tags with functionally equivalent tags that produce a more readable, maintainable, or portable HTML document. A similar caveat applies to HTML files created by a conversion service provider.

4.3 Testing

Whether created or converted, using an editor or a converter, documents need to be tested

before they are published, and whenever they are updated. Testing applies to both the publishing and the programming natures of electronic documents. Like in traditional publishing, testing includes: editing, proof-reading, spell-checks, grammar checks, punctuation, consistency, and formatting. Like with software, testing also includes executing the system: traversing the links (to the document, from the document, within the document), checking the user interface (how it looks on the screen), checking that all the pieces are available and reachable, and that downloadable versions are intact. These two natures can also be thought of as static and dynamic testing. Static testing is like desk-checking or code reading. Dynamic testing requires use of the hardware.

Development of procedures for testing and quality control of Web pages is a Web site management responsibility. The procedures defined do not have to be rigid or complex, but the testing requirements need to be considered, and budgeted for in both time and effort resource allocations. Codifying a test procedure also provides a way to capitalize on lessons learned, for example, the discovery that it is more efficient to spell-check the text of Web documents *before* adding HTML tags than afterward.

Static Testing
From reading E-mail messages and newsgroup postings, it is obvious that many people just type and send, without even reading what they've written, much less spell-checking it. That may be sufficient for informal and quick communication, which is akin to conversation, where people make mistakes while speaking, or change their minds in the middle of sentences. But it is not sufficient for a showcase, top-level Web page, that introduces the world to an organization's capabilities. The proper amount of testing to do, therefore, depends on the attributes of the document being tested. Some of these are:
- The size and complexity of the document
- Its location within the kiosk
- Its expected useful lifetime

- The size and distribution of the intended audience (among platforms, browsers, and domains)
- Its relative popularity, as determined from access logs
- The reasons for publishing the document

The last point can be illustrated by two extremes. At one end, if a page is created to advertise products, services, or other expertise, it needs to make the best possible impression. The errors in the following paragraph, taken from an actual Web page, may lead a browsing user to wonder about the quality of the rest of the services provided at that site:

> "Welcome to the XXXX automated reply form. Simply click on the buttons corrisponding [sic] to the information you require. The request will then be fowarded [sic] to the Webmaster for the XXXX and he will reply to your query ASAP. If you have any additional coments [sic] or questions just fill in the area at the bottom of the page."

At the other extreme is Tim Berners-Lee's discussion of documentation for which time spent testing may not be justifiable:

> "...there is very much information which is for a fleeting moment in people's minds, or is hastily scribbled down on some file, and which may be important to posterity. It is better for this information to be available even in unpolished form than for it to be hidden out of embarrassment for its form. Before electronic technology, the effort of publishing was such that this information was never seen, and it was a waste, and considered an insult to one's readers, to publish something which was not of high quality" [Berners-Lee 94c].

He does, however, caution that "it is important to make it clear what the quality of a document is when making a reference to it, to avoid disappointment" [Berners-Lee 94c].

Dynamic Testing
An important component of HTML document testing, that does not have an equivalent in typical document production, is testing the links among individual parts of a document, and to other parts of the Web. This part of the testing process is analogous to integration testing in software, where individual pieces of code are combined and tested to determine if they work together properly. Users will be unable to read a document if its links are broken, which effectively wastes all the effort that was expended to create the hyperlinks and mark up the document files.

Questions to be answered in dynamic testing include:
- Do the pages link together as planned?
- Are the images and other non-textual files there? Are they legible? Usable?
- Does the browser interpret the formatting commands as intended? Or at least acceptably?
- Is the performance acceptable?

Individual pages and documents can be tested using a browser, by providing the path and filenames for the pages, instead of a URL. With Netscape, this is the "Open File" command under the "File" menu. With Mosaic, the command is "Open Local." Browsing a file this way allows the Web author to check the appearance of each page, and try out the hyperlinks before the document is made accessible to outside users on the Web. Just like software programmers spend time debugging code, the creation of and conversion to HTML files almost always requires some iteration.

Hyperlinks coded with relative pathnames, however, may not work properly when displaying a local file, because the browser will interpret the path in the link anchor reference in relation to the host computer's root directory, rather than relative to the top-level http directory on the server. These links, therefore, can only be tested after the document has been published on the Web.

Table 14: Tools for Testing HTML and Links

TOOL NAME	DESCRIPTION
fix-html.pl	A PERL script to legitimize old HTML files into SGML-abiding HTML.
validation service	A form which will check HTML documents for errors, according to the latest specification.
weblint	Forms-based interface that checks HTML code [Unipress 94].
htmlchek.awk	Checks for defects in HTML files [Churchyard 95].
Web-roaming robot	Guido van Rossum's knobot code in "Python" language.
verify_links	Checks the links in a document for links to nonexistent resources, such as pages that have moved.
Web Checker	James Pitkow's web checking robot.
ALIWEB	An efficient distributed index harvesting system [Koster 94a].

A useful convention, although discouraged by many organizations, is to label documents at this stage of development as "Under Construction," or "Experimental." Other tricks are to use a non-standard port for the Web files, making it unlikely that surfing users will stumble onto the site accidentally, or to set file permissions so that only the person doing the testing has read access to the pages. Once testing is complete, the port and/or the permissions are easily changed, and the files become accessible to users at the site's "official" URL. Technical details about network connections and ports can be found in the references listed in Appendix B.

A final step in the testing process would be to access the document from somewhere else, if possible. Alternatively, a colleague on a remote client could be enlisted to try out the document and report back on access and retrieval times, glitches, appearance, etc. Once the document is on-line and in use, feedback can be solicited from users as to what works and what doesn't.

Testing HTML Forms
The CGI scripts which process form information need to be tested like any other software, because they *are* software. Even if the scripts have been acquired from elsewhere rather than locally composed, they still need testing in the new environment, in conjunction with the local HTML forms.

In general, any functions built in to Web pages, whether a database interface, an imagemap, or an interface to a WAIS, Archie, or Veronica

server, need to be tested to ensure they perform as intended.

Testing Tools
Tools for testing the correctness of HTML markup include those for checking the syntax of the individual pages, and those for checking the links to other pages. A few such tools are listed in Table 14; others can be found in Appendix B.

Testing for Portability
It is important to recognize that documents will look different using different browsers. For example, Figure 29 shows how the DACS home page appears when using CERN's WWW Line Mode browser. The numbers in square brackets indicate the hypertext links. Compare this with the Netscape interpretation of the DACS home page, shown in Figure 11. The portability of Web pages can be tested by displaying them on several browser and platform combinations, to determine if the appearance and/or legibility is acceptable on each. If a document is being developed for a specific browser, it need only be tested on that browser, but the browser for which the page has been designed needs to be identified somewhere on the document, e.g., "Netscape Enhanced," to alert users who may have different browsers.

4.4 Internal Review

As when developing large software systems, the next step after debugging and individually testing a document is often to pass it on for review by someone else. Since putting a document on the Internet is an act of publishing, documents posted on the Web are

```
┌─────────────────────────────────────────────────────────────┐
│ ▽                          native                             │
├─────────────────────────────────────────────────────────────┤
│ Welcome to the Data & Analysis Center for Software            │
│                                                               │
│ The DACS is a Department of Defense (DoD) Information Analysis │
│ Center                                                        │
│ (IAC[1]), administratively managed by the Defense Technical   │
│ Information                                                    │
│ Center (DTIC[2]) under the DoD IAC Program. The DACS is       │
│ technically                                                   │
│ managed by Rome Laboratory[3] (RL). Kaman Sciences            │
│ Corporation manages and                                       │
│ operates the DACS, serving as a centralized source for        │
│ current, readily                                              │
│ available data and information concerning software            │
│ engineering and software                                      │
│ technology.                                                   │
│                                                               │
│                                                               │
│ T                                                             │
│   Interested in a specific topic? Try searching the DACS      │
│   pages.[4]                                                    │
│                                                               │
│   About the DACS[5]                                           │
│                                                               │
│         DACS Information Overview                             │
│                                                               │
│         Spring 1995 Users Catalog                            │
│                                                               │
│         Initating a DACS Special Study                        │
│                                                               │
│ 1-11, Up, <RETURN> for more, Quit, or Help: █                 │
└─────────────────────────────────────────────────────────────┘
```

Figure 29: DACS Home Page Displayed by CERN's WWW Line Mode Browser

subject to the same review requirements as any other document that an organization publishes. Publishing a document electronically does not eliminate the need to follow established quality control and review procedures. The characteristics of the electronic document, however, will determine how much of the standard review process makes sense. There are three categories of electronic documents that can be treated differently in terms of internal review policies:

• Works originally authored for the Internet
• Previously published works converted to HTML for electronic publication
• Informal communications, such as E-mail and newsgroup postings

Publishing on the Web potentially broadcasts the information to the entire world, so it is important to consider what the documents look like, what information they contain, and the impressions they may give about the author and the author's organization.

Some organizations have consciously chosen to remove review requirements from WWW documents, in favor of an unrestricted approach to Web publishing. At Sun Microsystems, for example, groups and

individuals have been posting documents and information without prior review. One result of allowing employees the freedom to publicize their work and interests is that geographically distributed groups and departments have become more aware of each others' activities. The belief is that this would not have happened if Sun had insisted on reviewing everything first, and therefore the benefits they have realized outweigh the risks that are associated with exposing unfiltered information on the Web. Every organization needs to perform an assessment of potential risks and benefits to determine the level of control that is appropriate for what they intend to publish on the Web.

If an organization has a cognizant/responsible review person or group, then their services can be employed for review of original works. Any written policies that cover publishing can be followed to the extent they make sense for an electronic document. If an organization has well-established procedures for publishing paper documents, this information will be familiar. However, the people doing the electronic publishing may not be aware of them, especially if their backgrounds are in technical

Data & Analysis Center for Software

or programming disciplines, rather than in public relations or document production.

The review process needs to be tailored to the type of material. Home pages and other Web pages that function as directories, indexes, pointers, etc., have different review requirements than the pages that make up a document. Other policies that cover publishing of paper documents may not make sense for electronic documents. Some examples of publishing-related policies that will need to be interpreted include labeling with the company name or logo in a consistent manner, insertion of a standard disclaimer, inclusion of copyright notices, and identification of trademarks. Also in this category are company policies regarding protection of proprietary or sensitive information, the creation of duplicate and archive copies for company files, and the dissemination of copies according to a standard distribution list. For example, a hardcopy version of an electronic document may be generated for the corporate records management department, or some official notification of its existence and publication may be recorded instead.

Even if there are no formal review procedures within the organization, adding a review phase to the electronic publishing process will increase the quality of the Web site. Elements to consider in a defining a review procedure include:

- Identification and selection of reviewers; whether to include volunteers
- Specific questions for reviewers to answer
- Establishment of both time limits and deadlines for the review
- Allocating reviewers' labor in the overall budget for the publishing effort
- Allowing enough time to incorporate reviewers' comments into the document

Documents posted within the scope of a particular project, effort or contract may be subject to specific publication review and approval procedures such as would be defined in the contract statement of work. For deliverables and products developed for the DACS and for DACS technical area tasks, for example, the following list of requirements applies:

- "All items published and/or furnished by the DACS shall reflect that the products were prepared in part or wholly, as the case may be, under the auspices of the DACS, a DoD Information Analysis Center.
- "Approval ... of all contractor prepared technical output prior to release is required.
- "New products and services are subject to the approval of the Laboratory Program Manager" [SOW 91].

When existing, previously published works are converted to HTML for publication on the Web, additional reviews of the content will not be needed. Requirements for further review of the HTML version of the document will therefore be limited to checking for compliance with any identification and labeling requirements. Testing of the converted works will already have revealed any conversion-induced errors.

Items posted on the Web that are of a transient or informal nature will probably not need formal approvals. A common practice is to include a disclaimer at the end of each posting. Two examples of these from the Web are:

- "My views, not the U's" [from someone at a university]
- "Disclaimer: This document in no way represents the University of Pennsylvania. All opinions and errors are mine alone. Meng Weng Wong, mengwong@seas.upenn .edu" [Wong 94]

If an organization has established guidelines covering informal postings, Web authors need to adhere to them. The DTIC guidelines in Appendix D, for example, express a concern for including personal or frivolous information on Government-owned resources, warning that a disclaimer is not sufficient.

5 PUBLISHING

After a document has been tested and reviewed, the next step is to actually publish it. Web publishing involves two distinct activities. The first, called posting, is putting the data on the Web so that it can be accessed by other users. More importantly, if the data is to be of any use to anybody at all, is the second activity: announcing its existence so people can find, read, and use it [Rees 94]. Analogous activities in software development are delivery and installation, which put the system in the hands of the users. In traditional publishing, the works are printed, distributed, then marketed in various ways: reviews, advertisements, and author tours.

Even if a Web site is entirely internal to an organization, its existence needs to be announced and promoted to the members of the organization who can benefit from its development.

5.1 Posting

The process of posting is straightforward, assuming the Web server site is already operational. Information on server installation and administration is beyond the scope of this handbook, however, a good introduction, *Primer on WWW Servers*, is available on the Web [Torkington 93]. Others are listed in Appendix B.

An alternative to maintaining a server for publishing is to make arrangements to publish documents on another server. A number of both commercial and non-commercial organizations now provide server space for small Web publishers. Information about purchasing space on someone else's server is available from NCSA [NCSA 94b].

Documents are posted on the Web by putting the marked-up files (files with a .html or .htm extension) into a directory that is accessible from the Web server. The files' access permissions need to be set so that outside users can see the files, but need to be read-only so

that no one can alter them. If the document includes HTML forms, the corresponding CGI script files are put in the server's cgi-bin directory. If the document includes images or links to other non-textual material, read-only copies of those files are also put in an accessible location.

Some of the newer Web authoring tools include functions to perform this step automatically. For example, NaviSoft's NaviPress tool updates all the links in a collection of files when the collection is "saved to the server," using the "Save As" option under its "File" menu [Dozier 95]. Similarly, Interleaf's Cyberleaf tool contains a "post web" function, which "automatically copies completed webs to the Webserver" [Interleaf 94].

5.2 Publicizing

The second step of the publication process involves releasing information about new documents to as many relevant places, both on and off the Web, as practical. The types of places to announce new Web documents or services include:

- What's New Lists
- Virtual or Meta-Libraries (hierarchical)
- Meta-Indexes (searchable)
- Subject Indexes
- Traditional Marketing channels, e.g., advertisements, press releases, brochures, etc.
- Other Places, e.g., newsgroups

Some examples for each category are presented in this section, skewed where appropriate for software engineering and software technology information, because that is the DACS' area of interest. For most of these it is important to note that submission to the listing does not automatically create links back to the document. Someone (the Webmaster) at the other site needs to monitor the incoming announcements and requests, and take the time to put a link to the document into his or her HTML files. Some

sites have a regular update schedule, others may only be maintained as the local Webmaster finds spare time to keep it current. Webmasters may also perform a selection function, deciding whether or not the information is appropriate for their lists.

What's New Lists
Both Netscape's Navigator browser and the Software Development Group (SDG) at NCSA for Mosaic maintain What's New Pages. The DACS maintains a software-related subset of the Mosaic list.

Often, it is not very difficult to just register with one of these What's New pages for a quick way to gain some publicity for a new site. For example, registering with the *Netscape Tell Us What's New Page* is as simple as filling in an on-line HTML form which asks some short questions such as:
- Type of site, e.g., commercial, government, military, etc.
- URL and URL Page Title
- Contact information: name and E-mail address
- A short description of the site to be publicized [Netscape 95]

Table 15 lists some What's New Pages.

Table 15: Sample What's New Lists

NAME	URL (current as of August 1995)
SDG at NCSA	http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/whats-new.html
Netscape	http://home.netscape.com/escapes/whats_new.html
DACS	http://www.utica.kaman.com/awareness/whats.new.html

Table 16: Sample Meta-Libraries

NAME	URL (current as of August 1995)
Whole Internet Catalog	http://nearnet.gnn.com/wic/index.html
Stanford's Yahoo library	http://www.yahoo.com/
WWW Virtual Library at CERN	http://www.w3.org/hypertext/DataSources/bySubject/Overview.html
MCC's EINet Galaxy	http://galaxy.einet.net/galaxy.html
GNA Meta-Library	http://uu-gna.mit.edu:8001/cgi-bin/meta

Table 17: Sample Meta-Indexes

NAME	URL (current as of August 1995)
CUI World Wide Web Catalog	http://cuiwww.unige.ch/w3catalog
ALIWEB	http://web.nexor.co.uk/aliweb/doc/aliweb.html
WebCrawler	http://webcrawler.com/
Lycos	http://www.lycos.com
World Wide Web Worm	http://www.cs.colorado.edu/home/mcbryan/WWWW.html

Data & Analysis Center for Software

Virtual or Meta-libraries (hierarchical)

Procedures for adding a document to a meta-library can be found by checking the library's page that describes how to propose new information for inclusion. With these libraries, it's up to them whether or not to include a link to a document. Examples of some meta-libraries are listed in Table 16.

Meta-indexes (searchable)

Meta-indexes are created and maintained by tools such as Web robots, which search the meta-libraries for relevant information to include in their databases. Table 17 lists some examples of meta-indexes.

If a document has been included in a meta-library, it will eventually be found by the meta-index search tools, but that may take a long time. A more immediate approach is to submit it for inclusion directly to the meta-index. Instructions on how to do this can be found on the submissions pages for the WebCrawler, ALIWEB, or Lycos.

For ALIWEB, for example, the procedure is as follows:

1. Write a description of the service in a standard format in a file on the Web
2. Tell ALIWEB about the description file
3. ALIWEB retrieves the description file and includes it in a searchable database
4. Any Web user can search the database

An example in the format for the description file is shown in Figure 30. The most common template-types are SITEINFO, SERVICE, ORGANIZATION, DOCUMENT and USER. The template type affects which type of naming field is required, e.g., SITEINFO uses Host-Name, DOCUMENT uses Title, etc. [Koster 94a].

The ALIWEB documentation also contains style guidelines for including information in the submission, such as:

- Indicate how up-to-date the information/document being indexed is
- Identify the author and/or organization, especially if it is relevant to the contents
- Don't include information of interest to a small locally concentrated group
- Also don't index a list of links to other places
- Don't index every page, a description of one high-level page that has links to the details is sufficient
- Provide context in the title as well as the description [Koster 94b]

Subject indexes

There are many subject indexes on the Internet. Indexes appropriate to the subject matter of a document can be found by using the meta-indexes. Once suitable target subject indexes have been identified, their maintainers can be contacted via E-mail for instructions on how to submit information to them.

The University of Michigan maintains a clearinghouse for subject-oriented Internet resource guides [Rosenfeld 94]. Table 18 lists some subject indexes specific to software engineering and software technology information. Table 19 contains a list of relevant subjects found in one of those indexes, the Whole Internet Catalogue.

Template-Type:	DOCUMENT (*affects what other fields are included*)
Title:	Archie Gateways
URI:	/archie.html (*or the full pathname, e.g., http://web.nexor.co.uk/resource.html*)
Description:	A list of Hypertext Archie Gateways in the web.
Keywords:	Archie, other keywords

Figure 30: An ALIWEB Description File

Table 18: Software Related Subject Indexes

NAME	URL (current as of August 1995)
DACS Virtual Library	http://utica.kaman.com/awareness/vlib.html (see Figure 17)
WWW Virtual Library in Software Engineering	http://rbse.jsc.nasa.gov/virt-lib/soft-eng.html
Michigan State University provides the Usenet newsgroup comp.software-eng	gopher://gopher.msu.edu:3441/1threaded%20comp. software-eng
Unified Computer Science (CS) Technical Report Index at Indiana University	http://www.cs.indiana.edu/cstr/search
The Whole Internet Catalogue has a category Technology, that contains the category Computing	http://nearnet.gnn.com/wic/index.html (see Table 19)

Table 19: Computing Technology Entries in the Whole Internet Catalogue

Artificial Intelligence	Artificial Life ONLINE
CERT (Computer Emergency Response Team)	CERT Security Advisories
Communications of the ACM	Compression and Archival Software Summary
Computational Science Education Project	Computer Science Paper Bibliography
Computer Science Tech Reports	Computer Security and Encryption
Free On-line Dictionary of Computing	Free Software Foundation
High Performance Computing and Communications	HP Calculator BBS
Information System for Advanced Academic Computing	INRIA Bibliography
The Jargon File	League for Programming Freedom
Microsoft Corporation	Multimedia
Neural Networking Collection	National Institute of Standards & Technology
Principia Cybernetica Web	Public UNIX Access
Repository of Machine Learning Databases and Domain Theories	Software for Theologians
Supernet	U.K. Virtual Reality Special Interest Group Archive
UNIXhelp for users	UNIX Booklist
UUNET FTP archives	Xanadu Project

Data & Analysis Center for Software

Traditional Marketing Channels

So much publishing effort is concentrated on the electronic media that another reliable form of publicity is often forgotten. This form, which has been in use long before the electronic word, is paper publications and advertisements. With the coming of age of technology, paper media are often obscured by the magic of a medium like the Web.

Effective methods for publicizing a Web site are to simply send out a press release, or distribute a flashy paper flyer advertising the new site. This is a solid method of publicity because it allows Web authors to direct announcements to their target audiences. The idea here is to increase the total number of *quality* hits on the site rather than just the total volume number of hits. This is an important consideration, because it is quality hits which are needed to justify the development and maintenance costs of a Web site [Artner 95]. Press releases, brochures, annual reports, and flyer hand-outs are all fair game for generating Web site publicity.

Other places

Other on-line places to publicize Web documents include newsgroups, mailing lists and bulletin boards. The bulletin board service known as the "Mother-of-all BBS" includes a category for home pages of research centers [McBryan 94].

For time-critical, short-lived documents, or information for which immediate feedback or reader responses are sought, newsgroup announcements may be appropriate. When posting announcements of the existence and location of a document to USENET newsgroups and mailing lists, it is important to stay within the bounds of the groups' interest areas, especially if the announcement is commercial, or appears commercial. The purpose of the announcement is to generate interest, not annoyance, among the participants in the group. An example of a Web-related mailing list is www-announce.

Table 20 lists some computer-related (non-hardware) newsgroups found in the Whole Internet Catalogue.

Table 20: Comp. Newsgroups

ai	arch	cog-eng	compilers
compression	databases	dcom	editors
graphics	human-factors	infosystems.www	lang
lsi	multimedia	music	parallel
programming	protocols	realtime	research
robotics	security	simulation	software-eng
specification	terminals	theory	windows

6 OPERATION AND MAINTENANCE

Information providers have always been faced with the need to keep published information current. Traditionally, publishers have issued periodic updates of works to incorporate new and revised material. Textbooks, for example, may run to ten or more editions. Almanacs and similar time-sensitive reference books are typically revised and reprinted annually. Richard Bolles' classic job-hunting manual *What Color is Your Parachute?* has been updated and re-issued every year since 1975 [Bolles 92].

Software developers (and users) are acutely aware of the need to repair, revise, and enhance existing programs. These activities, known as software maintenance, consume as much or more effort than the original program development, because they continue throughout the entire useful lifetime of the software.

Electronic publishing, as a combination of publishing and software, exacerbates the problem of keeping information current. Because electronic documents are "easy to change" the rate of change expected by users is consequently higher. The volatility of the Internet, the explosive growth of the World Wide Web in particular, and its uncontrolled nature, all contribute to the maintenance effort required. The development of and adherence to maintenance guidelines is a highly recommended approach to dealing with these effects.

Design tradeoff decisions made during the development of hypertext documents affect the amount of effort that will be needed to maintain them. Similarly, some implementation styles and techniques will enhance document maintainability, while others may decrease maintainability. These ideas have been presented in the course of discussing the design and implementation of WWW information kiosks and documents. This section necessarily reiterates those ideas, but in the context of actually performing maintenance activities.

6.1 Web Site Maintenance

Operational concerns at the site level include:
- Resource allocation and consumption
- Internal user training, for both readers (browsers) and writers (authors)
- Monitoring accesses
- Evaluation: defining and applying criteria to gauge success

Maintenance concerns at the site level include:
- Organization: keeping relationships among individual documents coherent
- Consistency: maintaining a common look and feel among documents within the site
- Cost-effectiveness: applying resources judiciously for maximum benefit (congruent with goals)
- Technical Currency: staying abreast of technical developments, and incorporating those that improve the site (again, in terms of publishing goals)
- Content Currency: adding, updating, and removing documents as the timeliness of their information content warrants

In many ways, operations and maintenance are the same activity. A quality information space can not be operated for long without something having to be changed.

6.1.1 Structural Integrity

Before getting lost in the details of updating individual files, attention must be given to the affects of maintenance on the logical design of the site. The original design of the top-level page and the grouping of documents and sets of documents below it was intended to present a coherent, easily navigable information space to users. Additions and updates to the site must therefore fit into the established scheme, otherwise they can interfere with users' comprehension of the site. Also, considering the frequency with which corporations and other groups reorganize themselves, a Web site

Table 21: HTML Log Analysis Tools

TOOL NAME	DESCRIPTION
httpd-analyse.c	A program that changes the numeric Internet node numbers into domain names, thus 192.73.45.113 becomes utica.kaman.com.
WebStat	Package (written in Python, which is also required to run it) which supplies statistics on usage by domain, country, etc., with daily, weekly, monthly and annual reports available.
Wusage	C program which generates simple weekly reports in HTML, with inlined image graphs displaying server growth and the distribution of accesses by continent. Allows exclusion of irrelevant accesses, such as inlined images, and local machines, from the results.
Getstats	Server log analysis program, written in C, which provides reports for various time periods, with a high degree of flexibility. Add-ons include generators for reports in HTML and graphs [Hughes 94].
Getsites.c	Program which generates reports on a weekly or monthly basis.
wwwstat	Log analyzer written in PERL (used by the DACS Webmaster).

design based on internal administrative structures may not be the best choice for overall longevity.

A software-related illustration of this comes from early programmers, when compilers, such as FORTRAN and BASIC, required that the lines of code be numbered sequentially. As the programs written in these languages became more complex, and more frequently subject to maintenance updates and enhancements, programmers soon learned to number their lines by hundreds or even thousands, thus allowing "space" for later additions of code, without having to renumber the rest of the lines in the program. With a Web information site, the goal is to not have to reorganize everything to accommodate the addition of new information and new types of information to the site.

Another factor, as the site grows, concerns storage space on the server. Maintenance activities will need to include the removal of information as well as the addition of information. A software rule of thumb for newly delivered systems is to have up to 50 percent reserve capacity in the expected memory and processing resources, because of the tendency of delivered systems to grow. Allowing a similar capacity for growth and expansion will allow a Web site to adapt to increased demands, with less disruption than outgrowing the hardware would cause.

6.1.2 Access Monitoring

Keeping track of how users access documents will help a site focus its maintenance efforts. Active mechanisms, such as including an E-mail address or comment form in the document, can provide some feedback, but they require the cooperation of the users, as well as effort from the information provider to read and respond to them. Passive mechanisms, such as server logs, gather information without any intervention. To be useful, however, the information in the log needs to be reviewed and analyzed.

Logging tools can be used for tracking accesses to and transfers of data from Web documents. Log analysis tools can be used to help analyze and interpret the activity captured in the server log. Analysis is required to be able to distinguish quality hits from total hits, which feeds into the evaluation process. Table 21 lists several log analysis tools. More analysis tools can be found in the references listed in Appendix B.

The wwwstat tool, for example, provides the following counts and statistics:
- Percent of Requests (%Reqs)
- Percent of Bytes (%Byte)
- Total Bytes Sent (Bytes Sent)
- Total Requests (Requests)

Data & Analysis Center for Software

For each of the following categories:
- Daily Transmissions
- Hourly Transmissions
- Total Transfers by Client Domain
- Total Transfers by Reversed Sub-domain
- Total Transfers from each Archive Section
- Total Transfers to each Remote Identifier

Analysis of wwwstat tool server error statistics can also be used to determine when external links need to be updated. If the log shows an increase in the number of errors, that may imply that the URLs for one or more external links have changed.

In contractor-customer situations, or for internal systems, providing tools to the customers/users will allow them to communicate desired changes to the contractor/Webmaster. A prototyping tool, for example, would allow users to create mockups of how they would like the pages to look, or what specific information to include on them.

6.1.3 Configuration Management

Just like with computer software systems, an important component of maintenance is configuration management. As documents are maintained, the changes need to be tracked. Configuration management has three components: identification, change control and status accounting. Information about the set of documents maintained on a Web server that needs to be tracked includes where they are, when they were updated, and how recently their external links were validated.

Software engineers have evolved two strategies for addressing software configuration management needs: 1) establishment of well-defined change control procedures and 2) development of specialized configuration management tools. Procedures are defined and used to fill the gaps in tool support. At this stage of WWW development, however, no automated Web configuration management tools exist.

At the site level, controlling change requires controlling access permissions to the Web document files. Within an organization, it may be useful to restrict write permissions for files on the Web server to a single entity, such as the Webmaster. One reason for funneling documents through a single control point is to increase consistency among them, and enforce any organizational policies concerning content or format. Another reason is to prevent uncoordinated updates to existing documents that may have far-reaching effects and increase overall maintenance requirements. An existing configuration management control tool, such as the UNIX Source Code Control System (SCCS) could be adapted for this use.

Suggested tools to assist with tracking changes to individual Web documents could be as simple as a CGI script that reports the size of a file and the number of images attached to each link. Many word processing and desktop publishing systems have change notification and version identification features that could be adapted for this context.

When it is necessary to maintain consistency among several versions of a document -- the source files (in ASCII, graphics or word processor formats), their HTML equivalents for on-line browsing, and possibly PostScript and sequential ASCII versions for downloading -- configuration management becomes especially important. Maintainers need to be careful not make changes to one format without changing the other(s) to match.

The choices are to either edit in one format, then reconvert it to (all) the others, or to make the same edits in (every) version. It depends on how completely each conversion tool transforms a particular document into its target format, and how much they have to be touched up by hand after the conversion tool is finished. If the translation or conversion tools are comprehensive enough, it may be possible to set up a procedure whereby documents are created and updated only in their original "source" format. Then updates to the HTML, PostScript, or other formats could be accomplished by running the updated source file through the

various converters, rather than by editing the HTML, etc. documents directly.

As HTML conversion technology advances, this is likely to become the preferred approach. Cyberleaf, for example, allows documents to be revised using their original application, because the tool both matches the word processors' styles and uses a persistent link technology to automatically update and preserve existing hyperlinks [Interleaf 94].

If maintenance updates only affect the HTML version of a document, however, such as in the URLs of external links, the HTML could be edited independently of the downloadable version(s). This would need to be noted, however, in order to explain the differences in the file update times in the server directories. HTML comment tags are an appropriate place to record such updates.

6.2 Document Maintenance

Maintenance concerns at the document (page and file) level include content and structure, testing and tools. To maintain the highest degree of usefulness, the information in on-line documents needs to be updated regularly. Providing a list of points of contact, for example, is not very helpful if half of them have moved, or been reassigned. It is a waste of space and users' time to announce "upcoming" events that came and went six months ago.

Structure maintenance involves testing and updating the hyperlinks, both internal and external to the document. One consequence of updating documents is to consider the effects of changes from the point of view of outside users. Arbitrarily changing the URLs of existing documents will break any links to them that outside users have embedded in their documents, or may make it difficult for users to find the material again.

6.2.1 Updating Web Pages

Some types of information are more volatile than others. This will have been considered in

the original design of the documents, and will be reflected in the way the information has been decomposed into files and connected together by links. Documents that need to be changed often should be proportionally easy to change.

A related issue to be considered is the timing of revisions to on-line documents, due to the psychological impact that constant change has on readers. Basically, humans have problems relying on things that are not stable for some period of time. Therefore, if a revision is due out every day for a month, that fact could be noted somewhere in the document. Some users will choose to ignore the information until it is stable. Others, however, may want to watch it day by day. Many Web page developers use "Under Construction" and "Experimental" flags to indicate highly unstable material. Some Web information providers prefer not to publish the information at all until it has stabilized (for example, see the DTIC Guidelines in Appendix D). It is also helpful to include the date of last update in each file, either explicitly on the page, or in a non-displayed comment within the HTML file.

Added and updated Web pages need to undergo the same degree of testing as new documents. In addition to testing the links *from* an added/changed page, links *to* the updated page from other documents on a server need to be checked. This is similar in concept to the idea of regression testing in a software maintenance environment. Regression testing is a method for detecting errors spawned by changes or corrections made during software maintenance. A set of tests which the program has executed correctly is rerun after each set of changes is completed. If no errors occur, confidence is increased that no errors were spawned by the changes [Gloss-Soler 79]. It is especially important to test the behavior of the navigation aids, such as previous and next buttons, on the updated pages.

The same tools used to create HTML files can be used to maintain them. Which tool to use, however, depends on the extent of the changes that are being made to the files. When making minor updates to the information content of an existing Web document, the various HTML

Data & Analysis Center for Software

editors are more appropriate than the converters. Because the process of converting a document to HTML is usually iterative, maintaining the HTML version with an HTML editor will save steps over updating the source file, then re-converting it to HTML. When the changes to a document are extensive, however, a converter will probably be more efficient overall.

6.2.2 Maintaining Link Information

Web documents can have four types of hyperlinks, which have different maintenance implications:

- Links within the same file
- Links to other files on the local server
- Links to other locations on the Internet
- Links from other locations on the Internet

Maintenance of the first two types of links is more easily managed than the third. And although the fourth type of link is completely beyond the control of Web authors – there is no way of knowing the locations or extent of incoming external links -- their likely existence needs to be considered during maintenance activities.

Web document maintainers have access to information about the directory structure on their servers as well as some control over it, so maintaining consistency among the set of documents at a site is not an insolvable problem. Depending on the size of a local site and the number of people working independently on the same system, it may be necessary to set up a means of informing each other when one person's changes may affect other people's links. As documents get added or changed, or files moved around within the host computer's directory structure, the links embedded in the files need to be updated, so that they still work as intended. Alternatively, if files are on a UNIX system, symbolic links can be created between old file pathnames and their new locations. On a Macintosh, symbolic links can be achieved by use of the "File Alias" function, selectable from the "File" menu.

Changes in the third type of link can not be controlled, but have to be dealt with and therefore must be anticipated so their impact can be managed. One recommended approach for managing the maintenance of external link information is analogous to the programming technique of using labels for constant values. The current values of the constants are defined
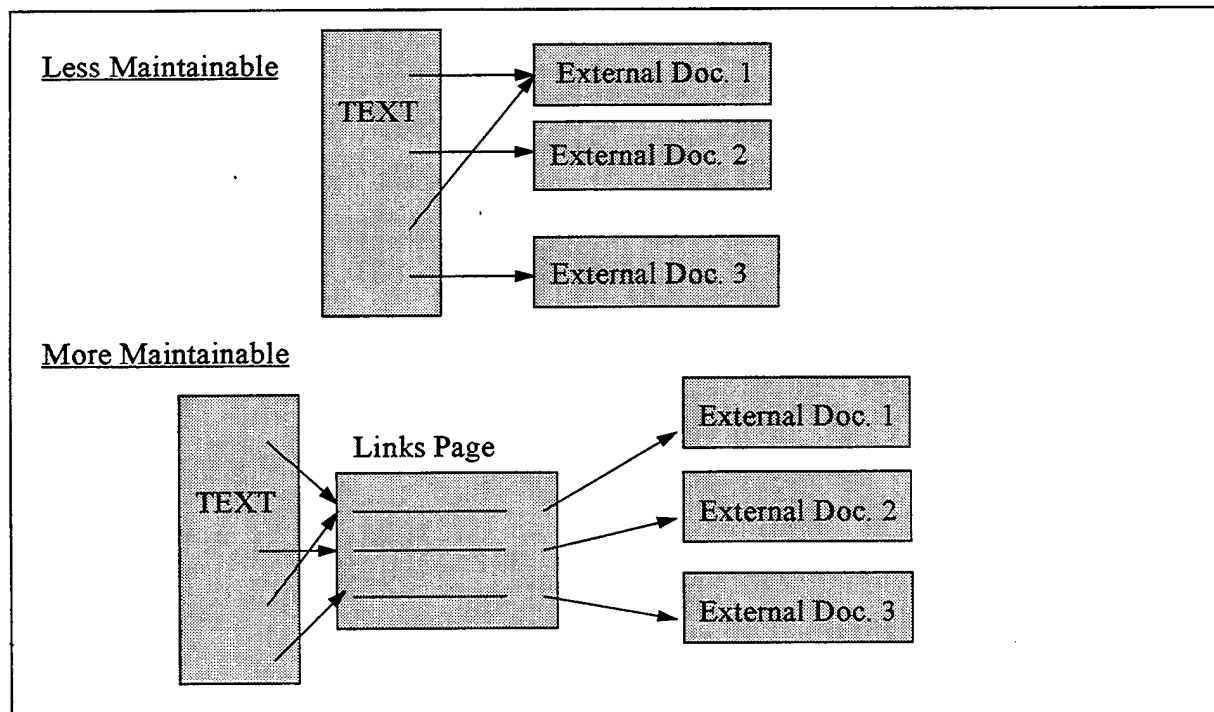
Figure 31: Aggregating External Link References

in a single place, e.g., in a header file. Everywhere else in the program that the constant is needed, the label is used instead. Then, if the value of the constant changes, the program only has to be changed in one place. For HTML documents, the "constant" is the URL of the external resource. The HTML equivalent of a label for the URL can be created by making internal links (using the NAME tag) to a single location in the document where the external links are coded. This approach is diagrammed in Figure 31. An application of this technique is described in Section 3.3.3, in the discussion of coding links to reference materials

It could be argued that putting this type of intermediate layer between the source and destination of a link counteracts the intent of the WWW to provide a feeling of seamless interconnection between diverse resources. But, people used to argue that structured programming limits the creativity of programmers. In evaluating the tradeoff between unrestrained creativity and maintenance effort, however, structured programming won. For hypertext documents that will have a significant maintenance lifetime, the same tradeoff is easily justifiable.

Inserting a link page between the link in the text and the external source will also create a performance penalty for the user, who must endure two retrievals when executing a link, one to retrieve the external links page and a second to retrieve the remote file. Some ways to minimize this penalty include:
- Putting the links page in the same file as the rest of the document, if possible
- Keeping the links page small
- Excluding non-textual material from the links page
- Grouping large numbers of external links onto several links pages, to minimize the retrieval time for each one
- Providing information about the size of external files and the number of images they contain on the links page

The fourth type of link is the inverse of the third type. Rather than links to other locations, it is other locations' links to this location and document. Anyone who has been frustrated by missing links when navigating the Internet knows how important it is to consider the effects on the outside users when updating a document.

Document Title: World Wide Web Frequently Asked Questions

Document URL: http://siva.cshl.org/~boutell/www_faq.html

World Wide Web Frequently Asked Questions

THIS DOCUMENT HAS MOVED

World Wide Web Frequently Asked Questions now resides on sunsite.unc.edu (at the URL http://sunsite.unc.edu/boutell/faq/www_faq.html). PLEASE update the page that brought you to this notice, or notify the maintainer of that page of the change. Thanks!

Many thanks to Sunsite for the space.

Tom Boutell

Back Forward Home Reload Open... Save As... Clone New Window Close Window

Figure 32: Announcement of Updated URL

One technique is to put a link to the newer version of a document in the old one. This way, when users try to access the old document, they will find out about the newer one, and can update their links accordingly. The real-world equivalents of this are the yellow forwarding address stickers the Post Office uses, and the "This number has been changed" message that telephone companies provide.

If the document is no longer "suitable for publication," or there is a need to conserve space on the server, the old page can be moved to an archive directory, or deleted from the system. In its place, however, a new page can be put at the old URL that contains a link to the location of the new version, and/or explains why the old contents are no longer available, to ensure that other people's links still work. An example of this is shown in Figure 32.

Relative vs. absolute pathnames
The choices of whether to use relative pathnames or absolute (full) pathnames in creating link references among documents and parts of documents on a server were made during the HTML implementation, but have an effect on maintenance efforts. A rule of thumb for minimizing the effort required to keep link references correct is to use relative pathnames for intra-document links (between parts of a single document), and full pathnames for inter-document links (between different documents). The rationale for this is explained in *A Beginner's Guide to HTML*, "consider a group of documents that comprise a user manual. Links within this group should be relative links. Links to other documents (perhaps a reference to related software) should use full path names. This way, if you move the user manual to a different directory, none of the links would have to be updated" [NCSA 94a].

A related implementation decision that has maintenance effects is in the level of detail attached to document navigation aids. For example, "Up" is less descriptive than "Up to Quarterly Report Table of Contents," but if the page is later incorporated in the Annual Report, the button labels would need to be changed to point up to the new location. If filenames and titles for Web pages are descriptive enough the

simple "Up" label on the navigation buttons will be sufficient.

Aliases

Use of an alias for the name of a server rather than the actual machine name is another fundamental practice for improving maintainability. If the server is moved to a different piece of hardware, redefining the alias is the only update needed. No link references need to be changed, and outside users need not even be aware of the move. (Although they may notice a difference in performance.) The alias www.host.type is becoming a popular convention for naming Web servers. This approach is similar to the layering schemes for defining network protocols, that shield implementation details from users, and thus increase portability.

Similarly, use of the default ports for the basic Internet protocols: 80 for http, 70 for gopher, 8000 for wais, increases maintainability. First, the URLs are simpler, because the port does not need to be specified. Second, it reduces the need to update the URLs of links as the network environment changes.

Tools for Maintaining Hypertext Links

Several HTML tools have been developed to help maintain links among HTML documents. Updated links can be tested with tools such as **html_analyzer** and the **EIT-Link-Verifier-Robot/0.2.** The readme file for **html_analyzer,** for example, explains:

"The intent of the html_analyzer is to assist the maintenance of HyperText MarkUp Language databases. As the number of HTML databases increases, the potential for hyperlinks that point to files or servers that no longer exist also increases. This results in the need for an automated hyperlink validation program. This is exactly what the html_analyzer does. The program also explores the relationship between hyperlinks and the contents of the hyperlink" [Pitkow 94].

The announcement for the **EIT-Link-Verifier-Robot/0.2** describes what it does as:

"...a link verifier to assist server maintenance. The link verifier tool starts from a given URL and traverses links outward, subject to a specified search profile, producing a report on the state of all the discovered links. The tool is invoked via CGI scripts, from your favorite browser" [McGuire 94].

Another tool, **verify_links**, checks the links in one document for links to non-existent resources [EIT 94].

Regular use of tools such as these can help Web maintainers verify the continued correctness of external links, whether or not any changes have been made to the local HTML files.

7 ISSUES

A number of open issues surround the distribution of information through the Web and the Internet in general. Most of the uncertainties can be classed as questions of either legality or control. The United States legal system has been unable to keep pace with the changes resulting from the rapid expansion and metamorphosis of the Internet. Many of the control issues are a result of holes in the technology, while others are related to the unknown impacts that increasingly pervasive electronic resources are having on society.

The purpose of this section is primarily to alert Web information providers to these open areas. Within each topic, pointers are provided to more detailed discussions of these issues. The following areas related to electronic publishing are included:
- Intellectual Property Rights
- Security
- Commercialization
- Standards

These issues are also interrelated. Increased commercialization of the Internet increases concerns about protection of intellectual property. Worries about potential copyright infringement fuel concerns about security. The need to achieve a return on investment drives efforts in standardization, which will decrease the costs of developing and maintaining a WWW presence.

7.1 Intellectual Property

The tradition on the Internet has been to freely trade information, generally not paying too much attention to intellectual property issues.

The doctrine of "fair use" provides some justification for these practices. Non-commercial use of portions of a work that does not harm the market can be considered fair use. Use of factual data is more likely to be considered fair use than use of artistic works or entertainment [Samuelson 94a]. As the Internet becomes more commercial, however, the fair use provision will be less widely applicable.

Four mechanisms exist in United States law for protecting intellectual property: patents, trade secrets, trademarks and copyrights [Jakes 89]. Of these, copyrights are the most relevant form of intellectual property for Web pages, although the possibility of infringing on an existing trademark must be considered in the choice of domain names [Quittner 94] and directory or filenames that become part of a URL. Trademarks used in hypermedia documents need to be identified, just as they are in any publication. Trade secrets and other proprietary information are not likely to be published on the WWW, but organizations may want to include such sensitive information on their internal network pages.

An original work is copyrightable whenever it is fixed in a tangible medium of expression. Copyrights protect the "expression" of an idea, but not the idea itself, although the line between the expression and the idea is often a matter of litigation. Works in which the idea and its expression are fused receive less protection under the law. Any right not expressly granted in a copyright notice, e.g., duplication, republication, etc., is by default retained by the copyright owner.

Figure 33: Sample Copyright Notices

Web information providers are faced with intellectual property issues in two directions:

- Providing protection for their own works
- Respecting protections on others' works

In the first direction, Web document authors who want to control how others use their data can protect their works by copyright. Although it is not necessary to include a copyright notice for a work to be protected, an explicit statement of what rights the author claims will help prevent others from infringing on them. Two suggested forms of copyright notices for use on the Internet are presented in Figure 33, which show examples of authors granting different rights.

In the second direction, authors of hypertext documents that include external links to pages developed by others need to be careful not to infringe on their copyrights. Developers of hypertext documents that include copies of portions of other works, such as graphics or images, need to be cautious when using unoriginal images, unless they are labeled "copyright-free," such as is often seen on collections of clip art images.

Some of the questions which arise from expanding traditional interpretations of intellectual property into a hypermedia environment include:

- Does the originator of included material need to be compensated?
- Does a Web author need explicit permission to include portions of another's work in a document?
- Must the owner of a work be notified whenever a link to that work is included in another document?

Even when the authoring organization clearly indicates on a document which rights are granted and which are reserved, actually obtaining permission to use the material or providing notification to the copyright holder may not be straightforward. For example, the first draft of this handbook included screen shots from a set of Web pages created for the 1994 World Cup Soccer tournament. The images were used in the handbook to illustrate Web site design. Later, traversal of a link to a copyright notice page revealed that use of those images without permission was prohibited. But the tournament is over, the organization no longer exists, and the telephone numbers given on the copyright page are not in service, so it was not possible to find anyone from whom to get permission to use those images. Consequently, they are not used in this version of the handbook. This anecdote also illustrates how it is necessary to maintain the information provided at Web sites.

Other nuances of the copyright law which have implications for Web document publishers include:

- Interpretation of terms such as "copy," "derivative work," and "modification" in an electronic media context
- Differences in the laws regarding different media and sources, such as music
- Inapplicability of United States law to international Web sites

Interpretation of existing intellectual property law with respect to computer applications is evolving. An organization's electronic publication process should therefore allow for legal consultation to obtain a current interpretation of the law on specific questions.

Otherwise, where the law is unclear, the best advice is to use common sense and courtesy.

Changes to the existing law are in process. A Working Group of the National Information Infrastructure Task Force (NIITF) has proposed expanding the copyright law in attempt to clarify digital transmission issues. Some argue that the proposed changes are too restrictive, to the point of effectively labeling browsing as an act of copyright infringement [Samuelson 94b][Wallich 95].

A new organization similar to the American Society of Composers, Authors, & Publishers (ASCAP) and the Copyright Clearance Center, Inc. (CCC), which act as clearinghouses for certain kinds of licenses and administer royalty payments, is being formed. Called the Authors Registry, the organization intends to maintain a database of author and agent contact information, monitor the use of registered authors' materials in electronic formats, including both on-line and CD-ROMs, and streamline royalty payment and accounting procedures for multimedia publications [O'Brien 95]. Also, the Copyright Clearance Center has recently established a WWW site that provides both on-line access to catalogs of royalty information, and a mechanism for permission-granting and reporting of photocopying copyrighted materials [Davis 95].

The Association for Computing Machinery (ACM) has defined a policy concerning copyrights that fits within its overall plans for transitioning from the production and distribution of traditional printed journals toward a comprehensive electronic publishing program. A complete description of the approach is contained in the April 1995 issue of the *Communications of the ACM* [Denning 95].

7.2 Security

Security issues and developments can be viewed from two perspectives. For information providers, the primary security issues concern protecting their information from unauthorized access and corruption, and receiving appropriate compensation for information products and services. From a user's point of view, the primary security concerns are protecting their privacy, avoiding fraud, and receiving products and services whose value is commensurate with their cost. In order to develop and maintain the confidence of their users, however, providers need to also include users' security concerns in developing Internet security strategies.

Some of the security concerns being expressed now will go away as the Internet becomes a more familiar place. Even without technological fixes and guarantees, people are devising work-arounds and procedures that mitigate some of the risks. The same phenomenon has occurred over and over in the evolution of financial transaction vehicles, from the acceptance of checks, to credit cards, to automated teller machines, to telemarketing and mail-order systems. Although there are still many instances of fraud, there are also well-known ways for consumers and providers to protect themselves, for example, never divulging a credit-card number during an incoming call.

Provider Security
Security-related issues for information providers include the protection of computers and computer-based information from unauthorized access and from either malicious or inadvertent corruption. Whenever a computer or Local Area Network (LAN) is connected to the Internet, security concerns arise because the potential for unwanted access or corruption has been created. As more commercial organizations connect themselves to the Internet, and as they increase their investments in and reliance on their Web information sites, the demand for security increases.

Many Web information providers protect their networks with firewalls. A firewall is a software mechanism that protects one trusted network from another, untrusted network. A typical scenario involving firewalls is where the protected network is a corporate enterprise network, and the untrusted network is the Internet. One portion of the firewall mechanism blocks traffic and another portion permits

traffic. Installing a Web server on the untrusted side of a firewall allows outside users to access the information on the server, although the server data is not totally secure and could possibly be modified by outsiders. Installation instructions for browsers usually discuss how to install them on the trusted side of a firewall so as to obtain data from servers on untrusted networks [Firewall 94]. More firewall information can be found in *Firewalls and Internet Security: Repelling the Wily Hacker* [Cheswick 94].

The software used to implement a WWW site, the server and/or the browsers, may themselves introduce security problems, of which installers need to be aware. For example, NCSA has identified three security problems in earlier versions of Mosaic for X Windows. Other browsers may contain other security flaws.

A security tool aimed at network administrators, called the Security Administrator Tool for Analyzing Networks (SATAN), can be used to probe a network for security weaknesses, which the administrator would then presumably fix. Critics, however, warn that the availability of this tool makes it much easier for hackers to invade other people's systems, because it can be used over the WWW [Wilder 95].

Several organizations are exploring security issues. For instance, the Rutgers University Network Services www-security team administers a mailing list devoted to WWW security issues. They also have a Web home page [Rutgers 94]. In addition, a number of security-related resources are available through the Internet, some of which specifically concern the WWW. A good index for security resources is maintained by the Web Project, Telecom Australia - Network Systems [TANSU 94]. Another resource is the Computer Emergency Response Team (CERT), which assists network communities in responding to emergency situations, identifying vulnerabilities, assessing systems, increasing user security awareness, and issuing advisories that detail security flaws found in hosts on the Internet [CERT].

Transaction Security

Other developments in security technology are being driven by the desire to conduct commercial transactions on the Web. It is becoming common for Web information providers to offer products for sale over the Internet to credit card buyers. But the communications take place over open channels, through which anybody could read the purchaser's credit card number. Most users desire a more secure transaction method before they will transmit private financial or personal information over the Internet. Another weakness of the Internet is the ease with which E-mail addresses and login IDs can be faked, because that raises doubts about how to ensure that orders are billed properly. The parties at both ends of a transmission involving a financial exchange need to be able to authenticate the transaction.

The technical infrastructure needed to support commercial transactions over the Internet is still being developed. Commercial versions of browsers are becoming available that have the features needed to support commercial transactions over the Internet. For instance, Netscape Communications Corporation's Netscape browser includes encryption capability. This functionality allows users to send confidential information, such as credit card numbers, securely. Netscape's commerce server includes authentication functions, which allow servers to ensure a message is really from the sending party [Patch 94]. As these capabilities become common, information products can be sold directly across the Internet with more confidence.

Transaction security is a very volatile area. As people are beginning to view the Internet as an inexpensive marketing and delivery system that is tied to a huge, untapped market, research efforts and resources are being focused on finding ways to exploit it safely and profitably. Several techniques are being tried. One involves use of software packages, such as SoftLock, which allow anyone to read portions of a document, but restrict access to the remainder. To purchase the rights to view the rest of the document, readers are directed to call an 800-number, where they will be given a

Data & Analysis Center for Software

password in exchange for their credit card number. Another requires buyers to pre-register by phone, at which time they are assigned a virtual credit card number to use for Internet purchases. Only the system knows the mapping between the virtual number and the real credit card number [CARI 95]. A third approach uses off-line communications to complete transfers from the users' "credit" accounts to the information providers' "checking" accounts [FV 95]. One thing all three of these approaches have in common is reliance on some non-Internet communication channel to verify or protect the sensitive information.

7.3 Commercialization

The Internet was not traditionally used for commercial purposes because the Acceptable Use Policy of the National Science Foundation (NSF), limited commercial use of the Internet. Since this policy is no longer enforced, the Internet is becoming increasingly commercial. Widespread commercial usage will change the culture of the Internet in directions that are not yet clear. Some suggest that the ratio of "newbies" to "netizens" is such that the previously existing Internet culture will be overwhelmed [Troetschel 95]. The influences and contributions of commercial sites and non-technical users will redefine Internet culture in ways that suit the new users and uses, which may not resemble at all the previous academic/researchy anarchy that characterized the pre-WWW, pre-commercial Internet.

For the near term, however, the Internet's non-commercial history and existing culture does affect the ability of Web developers to successfully implement commercial sites and services. Awareness of the acceptable ways of providing their information across the Internet will help avoid what existing users perceive to be misuse of Internet resources to advertise their services. The now infamous "Green Card" lawyers, for example, posted copies of an advertisement for their services to over 6,000 USENET newsgroups. When angry users flooded the lawyers' Internet service provider with complaints, the provider canceled their Internet access.

Organizations have success using the Internet as a commercial medium when they place their services within the existing culture. Intrusive methods of advertising, such as the lawyers' blanket newsgroup postings, are still considered unacceptable. Non-intrusive advertising can instead be achieved by developing a "presence," such as with a WWW site. The commercial advantages of a WWW site are twofold: the organization has control over the image it projects, and the information is available to users at their discretion. The WWW is thus a combination of interactive yellow pages and on-line ordering service, where a user can learn about a company's products and services, get ordering information, and do it all without the company having to send "junk" mail. As discussed in Section 5.2, commercial providers do need to publicize their Web sites, to let the Internet world know about their offerings on the WWW. Announcements about new services can be posted to the USENET newsgroup comp.infosystems.www.announce, and to newsgroups which are specifically related to the organization's area of interest.

Beyond publicity and advertising, organizations interested in offering information products for sale via the WWW need to consider how shipping information across the Internet will affect their public image. Organizational or contractual obligations regarding identification and marking of products need to be interpreted in the context of electronic delivery. For example, logos or distinctive bindings that are an integral part of the bound hardcopies of information products may not be translatable to electronic versions. When a report is printed by a user who has received it electronically, the provider can not control its physical appearance.

Until transaction security technology matures, the sale of information products through the Internet can not be completed reliably without some off-line communication. For now, the safe method of selling information products is to use the Internet as a promotional and awareness medium only. Information providers could include descriptions and abstracts of their available information products on their Web

sites, along with information about how to order the documents. The actual transactions, however, would use traditional channels. Payment and verification will still be by mail or phone.

Distribution and Delivery Issues
Some information products are limited in how widely they can be distributed, for example, documents may be proprietary, export controlled, or even classified. Providers of such information must consider the risks and potential liabilities of making it available on the Web. The problem is that once data is available on the Internet, it may shortly be transported to anywhere else in the world. Schemes for allowing access to only registered users can help, but can not guarantee complete control because once data is downloaded, it is trivial to retransmit it elsewhere. In the world of paper documents, this is also true: anyone can copy anything and send it anywhere. It is much faster and easier to copy and retransmit electronic documents, however. Domain names are no guarantee of a user's location, either. For example, a researcher in France might have an E-mail address at a university in the United States where he spent a recent sabbatical.

At a minimum information providers will need to clearly identify applicable restrictions on the links to restricted information and on the document pages themselves. For more protection, however, only unrestricted descriptions of limited distribution products would be put on the Web, along with information on how to obtain them. Ordering procedures for export controlled products may require that certain forms be filled out by the customer. The Web can provide these forms for users to fill out and transmit electronically. Thus, administrative procedures can be streamlined to take advantage of Web capabilities, while control over the actual distribution is maintained.

Pricing of electronic documents is another issue to be dealt with in developing a commercial Web information service. If prices of products are normally set to recover the costs incurred in their production and delivery, how are those costs calculated for electronic documents? A

considerable investment is involved in maintaining a Web site. The Webmaster's efforts in ensuring all information is current, addressing problems as they arise, and responding to feedback from users need to be funded. Yet as long as Internet charges remain connection-based, rather than transaction-based, the cost to download a document will be free.

Currently, efforts are being made to develop technologies in support of transaction-based pricing for some Internet services, but it is too early to know how this will turn out. In the meantime, organizations have several options to consider:

- Providing Internet-accessible products for free
- Allocating the cost of maintenance of the Web server among the documents available there
- If not, covering this cost some other way
- Relating the prices charged for electronic documents to the prices charged for hardcopy versions of the same documents

Once the pricing question has been answered, the charge for downloading could be indicated on each document. Payment could be accomplished through the establishment of subscription accounts, as is done with other commercial on-line services, such as CompuServe, America OnLine (AOL), and Dialog, or on an à la carte basis, as is done with Uncover and cable television's Pay-Per-View, once a user has established an account that can be billed.

Information Quality
Problems of data quality have arisen with the exponential growth of the Web. Information on the Web ranges from very poor to very high quality, like software products that range from quick and dirty routines, to freeware and shareware, to slick commercial packages and highly-reliable embedded software. The filtering and selectivity of information that development of Web pages entails is in some sense equivalent to the filtering that takes place in more traditional media, but it is not nearly as restrictive. Because of the ease with which information can be added to the Web, a wider

Data & Analysis Center for Software

range of both voices and quality are found on-line than in traditional media.

The emergence of electronic journals is an example of an area where information quality is being questioned. Electronic journals are becoming an alternative outlet for publicizing scientific research information. They have the advantages of being able to publish articles much sooner and of costing nearly nothing compared to their traditional paper journal counterparts. They lack, however, the prestige of the paper journals, partly because they lack a formal review process equivalent to the peer review system of traditional publications. Procedures for increasing the level of review are beginning to be employed for some electronic journals published on the Web. They also provide the capability for readers to add comments to an article [Odlyzko 94] [Leslie 94]. One journal requires two levels of review, and then includes up to 30 commentaries along with the actual article [Stix 94].

An organization can respond to quality of information issues by striving to engineer quality Web sites and documents. A consistent image in electronic documents will communicate a level of effort and quality that contrasts with hasty, temporary, and not necessarily trustworthy Web pages. For example, IACs are by definition the authoritative sources of information in their respective technical areas. The appearance of their Web pages needs to confirm rather than contradict that authority. Instead of relying on a distinctive binding or packaging for information products, they can develop the Web equivalent -- a distinctive look and feel for their Web pages. An IAC can also add value to its Web pages and provide assistance to Web navigators by annotating links to external resources in its technical area with additional information about the quality, technical relevance, and extent of the external resources. An example of providing information about what's at the other end of a link, from the DACS Virtual Library page, was shown in Figure 17.

Market Research
As in any commercial undertaking, having reliable, up-to-date information about customers and potential customers allows for more relevant product development and more focused promotion. On the WWW, some of the data needed for market research analysis can be collected automatically. Tracking accesses to Web pages allows the developers to gain insight into their success at attracting readers. It also shows what links are particularly interesting to users. Monitoring tools can capture the domains and subdomains of the clients that access a document. That allows the information providers to make inferences as to whether or not they are reaching their target audiences.

The effectiveness of promotion efforts can be gauged by correlating them to access rates of relevant pages. Accesses to the DACS home page, for example, increased 10-fold as soon as it appeared on NCSA's What's New list. This illustrates the importance of proper promotion of new Web services, as discussed in Section 5.2. Analysis of the domains of these new users indicated that many were probably just curiosity-seekers, not likely to be interested in the DACS' area of expertise, and not likely to become DACS users. Because the Web encourages browsing, the number of visits to a high-level page may not reflect deep interest in the topic. Files that are downloaded, however, probably indicate a high degree of interest. A distinction between total hits and quality hits therefore needs to be made when analyzing and interpreting access data. Another caution regarding Web server log data involves pages with embedded images. They are counted separately, so the number of accesses to a page with many images may artificially appear larger than the number of accesses to a page without images. Some of the more sophisticated log analysis tools are designed to distinguish between these, and therefore provide more accurate data.

7.4 Standardization

In the same way that the purpose of this handbook is to collect and present the current best practices for publishing Web documents,

the goal of standards developers is to define recommended or required practices. Standards are generally more rigorous than the guidelines in this handbook. They are used to codify best practices, to provide reference information, to establish order, and to make things easier for both developers (publishers) and users (readers). The emergence of standards is an indication that an engineering discipline is maturing [Shaw 90].

Information about standards is included in the handbook because the use of standards increases the portability and interoperability of products, and can decrease maintenance and development costs. Conforming to established standards provides a way to take advantage of the trial and error of predecessors, without incurring their learning costs. Many Web information providers, recognizing the benefits to be gained from standardization, are developing internal standards and guidelines, tailored to their own environments and requirements. An example, developed by DTIC, is included in Appendix D.

There is currently much standards development activity related to WWW publishing. It is important for authors to be aware of both current and evolving standards, in order to more easily maintain compliance with them. Knowledge of proposed changes will also help authors create documents that can later be more easily updated to take advantage of newer capabilities. It is important to recognize, too, that standards and guidelines themselves must be maintained to keep pace with changes in the technology.

7.4.1 Markup Standards

These standards and standards activities are related to the creation aspects of WWW publishing.

HTML
The hypertext markup language is not yet defined by a standard. The current definition of HTML includes features that are defined as either "official" or "unofficial." The official ones are supposed to be interpretable by any browser. The unofficial ones may work on some browsers, but not others, or may behave unpredictably [Berners-Lee 94c].

The group working to define a standard for HTML is the Internet Engineering Task Force (IETF). Its approach is to codify existing practices on the Internet, and define them so that development of new features is straightforward. The first specification will define HTML Level 2, which includes basic features, highlighting, images and forms. The HTML test pattern in Appendix E contains the tags included in HTML Level 2. The standard will also specify the relationships between HTML and other standards and practices, such as SGML.

The group's intent in developing a standard definition for HTML is to provide a mandatory common format for all World Wide Web applications. After completing the HTML2 standard definition, the IETF will work on defining the next levels of HTML, which are known as HTML+.

HTML+ (HTMLPlus)
HTML+ is the designation for features and capabilities that will be added to future levels of HTML. The tags and formatting capabilities proposed for inclusion in HTML Level 3 (HTML3) are currently under discussion. In the current draft, HTML3 will allow a gradual roll-over from the HTML2 formats, with features like tables, captioned figures and fill-out forms for querying remote databases or mailing questionnaires. The draft also includes a proposal to add support for mathematical formulas. Authors will be provided with some more controls over how documents are presented on client systems. Further controls will be possible through the proposal for incorporating style sheets [Raggett 95]. Some browsers already support some of the features proposed for HTML3, such as fill-out forms.

Further enhancements to HTML are referred to as HTML4 and up. It is expected that markup sets defined as the HTML+ levels will be strict supersets of current HTML, so that existing HTML documents will be completely readable by any HTML+ level [Berners-Lee 94c].

The dual goal of supporting a wide range of display types and keeping browser software as simple as possible limits the complexity of the markup that can be included in HTML. The disparate needs of authors has led to the inclusion of limited rendering hints. The features that are supported arise from several years experience with the World Wide Web and the existing HTML format [Connolly 94].

SGML
The Standard Generalized Markup Language (SGML) is an ISO Standard, ISO 8879. SGML addresses the need for capturing the logical elements of documents as opposed to the processing functions to be performed on those elements. SGML is essentially an extensible document description language, based on a notation for embedding tags into the body of a document's text. The markup structure permitted for each class (type) of documents is defined by an SGML Document Type Definition (DTD). The current version of HTML is SGML-compliant, but covers only a subset of SGML; HTML3, however, is being defined as a true SGML DTD.

HyTime
The Hypermedia Time-based Structuring Language (HyTime) is an extension of SGML into hypermedia and multimedia documents. The HyTime Language definition includes a query language for dynamically selecting components of a HyTime document based on both structural attributes and content [Buford 95]. HyTime is seen by some as a more flexible and more powerful document architecture for WWW publishing than HTML. HyTime is defined by ISO 10744 [ISO 92].

Text Encoding Initiative
The Text Encoding Initiative (TEI) is an international research project for SGML-based document exchange in the humanities. Since 1988, it has been working toward the definition of a suite of extensible Guidelines and Recommendations for use when encoding text in machine readable form for research purposes. Its initial proposals recommended the adoption of a standard based on SGML, and made very detailed proposals for document type definitions covering a large range of document types, including tagsets for basic prose, dictionaries, lexical and syntactic analyses and textual criticism.

Defacto Standards
As often happens in emerging technologies, widespread adoption of a particular tool or method compels the rest of the market or industry to conform. Adobe's PostScript is an example of a product that has become a defacto standard for printed documents. Apart from the official, standard definitions of HTML capabilities, therefore, the features supported by Netscape are seen by many as a defacto HTML standard, just as Mosaic's capabilities effectively defined HTML last year.

7.4.2 Retrieval and Application Standards

From a user's perspective, research into making access and retrieval of information from the Internet more efficient is as important as advances in authoring capabilities. This is especially true as the information available proliferates and the user community broadens. It matters little how well-constructed a Web site is, if no one knows about it. Although retrieval issues are outside the scope of this document, Web authors who are aware of information search and retrieval standards can use them to increase the usefulness of their Web documents. Following conventions for indexing and describing the information in a document will enable those who expect and rely on those conventions to find that document. The following paragraphs contain examples of these.

ANSI Z39.50
The "American National Standard Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection" is developed by the National Information Standards Organization (NISO), accredited to the American National Standards Institute (ANSI). ANSI Z39.50 complies with the Open Systems Interconnection (OSI) family of standards promulgated by the International Organization for Standardization (ISO), and is interoperable with the international standards

for information search and retrieval, ISO 10162 and 10163.

USMARC
USMARC is an implementation of ANSI/NISO Z39.2, the American National Standard for Bibliographic Information Interchange. The USMARC format documents contain the definitions and content designators for the fields that are to be carried in records structured according to Z39.2. GILS (see below) records in USMARC format contain fields defined in USMARC Format for Bibliographic Data. This documentation is published by the Library of Congress.

Government Information Locator Service (GILS)
The US Government is developing a standardized approach for identifying locations of government information on the Internet. The GILS effort is being coordinated by the US Geological Survey (USGS). The definition and description of GILS began in 1994 and will continue through 1995. A subset of the GILS core elements that are recommended for consideration in the implementation of home pages is shown in Figure 34 [GILS 94]. The goals of the GILS project are congruent with some of the needs identified in this handbook, as can be seen by comparing the contents of Figure 34 with the discussion of extended

document information described in Section 3.3.2, and illustrated in Figure 19. An example of a GILS-compliant Web page from DTIC is included in Appendix D, following the DTIC WWW publishing guidelines.

The goal of establishing an agency-based Government Information Locator Service is to help the public locate and access information throughout the Federal Government, as part of the Federal role in the National Information Infrastructure (NII) [Christian 94]. Sources of more GILS information are listed in Appendix B.

7.4.3 WWW Consortium

The WWW Consortium (W3C) is an outgrowth of an earlier joint effort between CERN and MIT, known as the WWW Organization (W3O), which was formed to augment and continue the Web development work begun at CERN. The W3O collaboration effort was redefined as a consortium to open up the activities for participation to any interested groups.

The purpose of the consortium is to further the work on technical implementation details needed to support new applications and uses

```
        Title
        Control Identifier
        Abstract
        Purpose
        Originator
        Availability
                Distributor
                Resource Description
                Order Process
                Technical Prerequisites
                Available Time Period
                Available Linkage
                Available Linkage Type
        Point of Contact for Further Information
        Date Last Modified
        Agency of Program
```
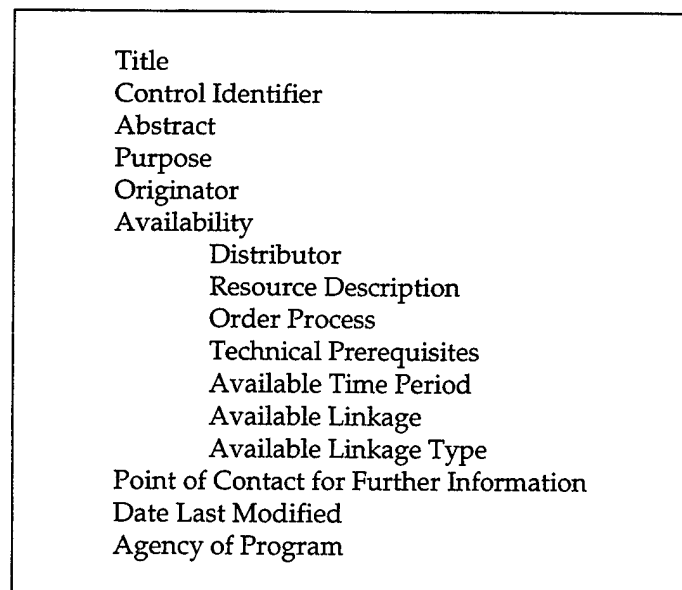
Figure 34: GILS Recommended Home Page Elements

Data & Analysis Center for Software

that continue to be defined for the WWW. The intent is not to compete with commercial activities in Web product development, but to guide the Web's evolution so that it moves toward interoperability and standardization, rather than toward proprietary or incompatible enhancements. By soliciting involvement from industry contributors in the development of new protocols and in standardization efforts, the consortium seeks broad acceptance of its products and approaches.

Four technical areas identified as initial priorities by the consortium are:
- Automation support for frequently used manual procedures
- Incremental additions of new information and object types to the Web
- Technical solutions to performance and scale issues caused by Web growth
- Solutions for user authentication, data integrity and privacy issues

These areas encompass work on designing new protocols, adding enhancements to the markup language, and coordinating Web standardization efforts with relevant standards from other organizations. A complete statement of the goals and objectives of the W3C, along with information on how to participate, is available in the W3C Prospectus [Berners-Lee 95].

8 CONCLUSIONS

Electronic publishing is different and new in several ways, and therefore has consequences, unknowns, technology holes, and social-cultural voids. But electronic publishing is also related to its antecedents, both in traditional publishing and in software development. The solution for dealing with the technologies and challenges of electronic publishing is to incorporate the best practices and lessons learned from its antecedents, adapting them where necessary.

Success in electronic publishing will come from understanding and enduring the risks, in order to achieve the potential benefits. To recall the basic analogy presented in this handbook, early difficulties with programming didn't stop society from becoming more and more software-dependent. Software developers and users endured punched cards, batch processing, and assembly language programming during its growth and evolution, even though the technology never quite keeps up with the public's expectations and desires, which grow faster. Processes for creating and managing software have evolved, too. It is now almost possible to say "software engineering" without having real engineers snicker or be offended.

An engineering approach is not just applying rules, but applying engineering judgment, which is achieved from an understanding of causes and effects. It is necessary to think about what is best in each situation, for example, with document page sizes, neither logical nor physical consistency is always better. Blindly consistent applications of some rule of thumb or standard will not be the best approach. Thus, Web information providers need to find the balance between discipline and flexibility.

The second component of success concerns the information content of the Web site. The importance of the initial requirements analysis steps, where the objectives for developing a WWW information site are determined, can not be over-emphasized. Part of defining a Web presence must include a review of what else is already available. Merely understanding one's own perspective is not enough, because many organizations may be developing Web sites devoted to similar topics. The creator of an authoritative source on Computer Mediated Communication (CMC), John December, hypothesized the following pattern in the development of information spaces on the Internet, including FTP, Telnet, Gopher, and the Web:

1. "Developers introduced an information presentation protocol or system.
2. "Users contributed information to the resulting information space, leading to:
   - Information space saturation - a plethora of information servers and an abundance of content. This abundance grows to such a degree that the space can't be encountered without information layering or filtering by way of hand-crafted indexes or other guides to the spaces.
   - Information space pollution - redundant, erroneous, or poorly maintained information becomes replicated throughout the space, obscuring other information.
3. "Developers created tools to automatically traverse the space and glean information about resources. The results of this automated gleaning is a database which can be queried through a keyword or other indexing scheme.
4. "With greater visibility of the available resources, redundancy decreased and specialization increased. Specialized information servers, often under the guidance of experts in the subject area of the information, created new levels and standards for quality. Often, lists or indexes of information servers also contribute greatly to this process (for example, the well-known Gopher Jewels showcase specialized Gophers, discouraging duplication and encouraging specialization)" [December 94].

The Web is somewhere in the third or fourth stage of this cycle. Developing and maintaining

a WWW presence that becomes a recognized source of authoritative information therefore requires:

- Domain knowledge
- A well thought-out organization of that information
- Work to determine what information is available, both off the Internet and on it
- Continuous maintenance
- Sufficient resources to accomplish these objectives

Evolution

Forecasts for the future of electronic publishing are many and varied, as can be seen from ideas presented at the Dartmouth Institute for Advanced Graduate Studies (DAGS) conference in June 1995 [DAGS 95]. An easy prediction to make is the availability of more powerful tools that move implementors away from dealing directly with HTML and technical details. Other predictions include the ability to convert documents to browser-interpretable formats on the fly; and an increase in database-based publishing, where documents are generated in response to user queries. Developments in retrieval technology will also have an influence on both authoring decisions and techniques.

Despite all that is expected to change with respect to technology and tool specifics, an engineering approach will give Web information providers a sound foundation. Much of the advice in this handbook can be extrapolated to whatever new implementation techniques arise.

APPENDIX A DEFINITIONS

A.1 Acronyms

4GL	Fourth Generation Language
ACM	Association for Computing Machinery
AFB	Air Force Base
AI	Artificial Intelligence
AIFF	Audio Interchange File Format *Apple Macintosh*
ALIWEB	Archie-Like Indexing for the Web
ANSI	American National Standards Institute
AOL	America OnLine
ASCAP	American Society of Composers, Authors, & Publishers
ASCII	American Standard Code for Information Interchange *plain text encoding*
AU	Audio file format *Sun Microsystems*
AWK	UNIX language
BBS	Bulletin Board System
CASE	Computer Aided Software Engineering
CCC	Copyright Clearance Center, Inc.
CD-ROM	Compact Disc Read Only Memory
CERN	European Particle Physics Laboratory
CERT	Computer Emergency Response Team
CGI	Common Gateway Interface
CMC	Computer Mediated Communication
CRT	Cathode Ray Tube
CS	Computer Science

CUI	Centre Universitaire d'Informatique
DACS	Data & Analysis Center for Software
DAGS	Dartmouth Institute for Advanced Graduate Studies
DID	Data Item Description
DoD	Department of Defense
DD	DoD Form Number System Indicator
DOS	Disk Operating System *Microsoft*
DTD	Document Type Description
DTIC	Defense Technical Information Center
DTIW	Defense Technical Information Web
EIT	Enterprise Integration Technologies
E-mail	Electronic Mail
FAQ	Frequently Asked Questions
FTP	File Transfer Protocol
GIF	Graphics Interchange Format *CompuServe*
GILS	Government Information Locator Service
GNA	Globewide Network Academy
GNU	GNU's Not UNIX *organization, part of Free Software Foundation; responsible for EMACS*
GOTO	Go To *command used in programming*
GUI	Graphical User Interface
GZ	GZIP's Compressed File Format
GZIP	GNU Project's compression program
HOL	Higher Order Language
HP	Hewlett Packard

HREF	HTML tag for links
HTM	Alternate Extension for HTML Files
HTML	Hypertext Markup Language
HTML+	Anticipated Upgrade of HTML *also called HTMLPlus*
HTTP	Hypertext Transfer Protocol
HyTime	Hypermedia Time-Based Structuring Language
IAC	Information Analysis Center
ID	Identification
IDEF	Integration Definition
IETF	Internet Engineering Task Force
IMG	HTML Tag for Inlined Images
INRIA	European Institute
ISDN	Integrated Services Digital Network *communications protocol*
ISO	International Organization for Standardization
JPEG, JPG	Joint Photographers Experts Group
KB, Kbytes	KiloBytes
KBS, Kbps	KiloBytes per second
kHZ	KiloHertz
LAN	Local Area Network
LaTeX	Typesetting Format/Language
MCC	Microelectronics and Computer Technology Corporation
MIF	FrameMaker File Format
MIT	Massachusetts Institute of Technology
MOV	Quicktime Movie Format *Apple*

MPEG, MPG	Motion Photography Experts Group
MS	Microsoft
NCSA	National Center for Supercomputing Applications *at the University of Illinois at Urbana-Champaign; invented Mosaic*
NII	National Information Infrastructure
NIITF	National Information Infrastructure Task Force
NISO	National Information Standards Organization
NIST	National Institute of Standards and Technology
NLM	National Library of Medicine
NNTP	Network News Transfer Protocol
nroff	UNIX Text Formatting
NSF	National Science Foundation
OCR	Optical Character Recognition
OSD	Office of the Secretary of Defense
OSI	Open Systems Interconnection
PC	Personal Computer
PCX	Paintbrush File Format *Microsoft*
PDF	Portable Document Format
PERL	Practical Extraction and Report Language UNIX language
PIXEL	Picture Element
PS	Postscript
RL	Rome Laboratory
RTF	Rich Text Format
SATAN	Security Administrator Tool for Analyzing Networks
SCCS	Source Code Control System

SDG	Software Development Group *organization at NCSA*
SED	UNIX language
SEI	Software Engineering Institute
SGML	Standard Generalized Markup Language *ISO Standard 8879*
SLIP/PPP	Communication Protocol(s) for connection to the Internet
SOAR	State of the Art Report
SRC	HTML tag for inlined image location
STI	Scientific and Technial Information
TANSU	Telecom Australia-Network Systems
T1	Type of phone line
TCL	Tool Command Language
TCP/IP	Transmission Control Protocol/Internet Protocol
TEI	Text Encoding Initiative
TIFF	Tagged Image File Format
TOC	Table of Contents
troff	UNIX Text Formatting
TXT	ASCII Text Format Designator
UIUC	University of Illinois at Urbana-Champaign
UK	United Kingdom
UNIX	Operating System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USENET	Internet Network of Newsgroups
USGS	United States Geological Survey
USMARC	United States Machine Readable Cataloguing

UUNET	UUCP Network
VRML	Virtual Reality Markup Language
W3	World Wide Web
W3C	WWW Consortium
W3O	WWW Organization
WAIS	Wide Area Information Server
WWW	World Wide Web *also known as the Web, W3*
WYSIWYG	What You See Is What You Get
XBM	X Windows Bit Map

Data & Analysis Center for Software

A.2 Glossary

This glossary is intended to provide a brief explanation of some of the terms used in this handbook. Rather than duplicate information available in other, more comprehensive sources, this list focuses on terms that have a specific meaning in the context of this handbook. Many of the HTML guides listed in Appendix B include glossaries. Matisse Enzel's Glossary of Internet Terms is a good list of definitions [Enzel 95].

Author <=> Implementor <=> Publisher <=> Information Provider

Authoring Creating or converting HTML files.

Conversion Adding HTML markup to an existing document.

Document Logically distinct set of pages - not all pages are part of a document.

Home Page Top-level page for a site, containing and organizing links to the information below

Kiosk <=> Site A WWW server location, and all the files accessible from it.

Page <=> File A separately retrievable unit of WWW information.

Reader <=> User

APPENDIX B SOURCES

B.1 References

[Adobe 94] at URL: http://www.adobe.com/Acrobat/Acrobat0.html, Cited October 28, 1994, 3190 bytes.

[Agresti 86] William W. Agresti, *New Paradigms for Software Development*, New York: IEEE Computer Society Press, 1986.

[Artner 95] Bob Artner, "The Revenge of the Press Release!" *Internet Business Advantage*, Vol. 1, No. 3, July 1995.

[Balasubramanian 94] V. Balasubramanian, *State of the Art Review on Hypermedia Issues and Applications*, Graduate School of Management, Rutgers University, Newark, New Jersey, March 1994, at URL: http://www.isg.sfu.ca/~duchier/misc/hypertext_review, Cited August 1995.

[Berners-Lee 94a] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, "The World-Wide Web," *Communications of the ACM*, V. 37, No. 8, August 1994, pp. 76-82.

[Berners-Lee 94b] Tim Berners-Lee, *W3 and HTMLTOOLS*, at URL: http://www.w3.org/hypertext /WWW/Tools/Overview.html, Cited October 21, 1994.

[Berners-Lee 94c] Tim Berners-Lee, *Style Guide for Online Hypertext*, at URL: http://www.w3.org /hypertext/WWW/Provider/Style/Overview.html, Cited October 24, 1994; 1642 bytes.

[Berners-Lee 95] Tim Berners-Lee, *World Wide Web: The Consortium and Plans for the Future*, Proceedings DAGS, June, 1995, p. 73. Also at URL: http://www.w3.org/hypertext/WWW /Consortium, Cited May 1995.

[Bolles 92] Richard Bolles, *What Color is Your Parachute?* 12th edition, Berkeley, CA: Ten Speed Press, 1992.

[Booch 86] Grady Booch, *Software Engineering with Ada, Second Edition,* Menlo Park, CA: Benjamin/Cummings Publishing Company, Inc., 1986.

[Boutell 95] Thomas Boutell, *World Wide Web Frequently Asked Questions,* at URL: http://sunsite.unc.edu/boutell/faq, Last Updated July 13, 1995.

[Buford 95] John F. Buford, *Evaluation of a Query Language for Structured Hypermedia Documents*, Proceedings DAGS, June, 1995, pp. 105-116.

[CARI 95] "CARI on, my Webward Businesses," *Internet Business Advantage*, V. 1, No. 2, June 1995, p.14.

[CERT] Computer Emergency Response Team advisories are available via anonymous FTP from cert.org.

[Cheswick 94] William R. Cheswick and Steven M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Reading, MA: Addison-Wesley, 1994.

[Christian 94] Eliot Christian, *GILS*, available on the FedWorld electronic bulletin board [703-321-8020] or via FTP at info.er.usgs.gov in the directory /public/gils as gils.doc [Microsoft Word for Windows 6.0 format], gils.rtf [Rich Text Format], or gils.txt [ASCII text format], Cited November 3, 1994.

[Churchyard 95] Henry Churchyard, *htmlchek - HTML Error Checker and Utilities*, Version 4.0, University of Texas at Austin, at URL: http://www.w3.org/hypertext/WWW/Tools/htmlchek.html, Updated January 23, 1995.

[Connolly 94] Daniel W. Connolly, *HyperText Markup Language (HTML): Working and Background Materials*, at URL: http://www.w3.org/hypertext/WWW/MarkUp/, Cited November 3, 1994.

[DAGS 95] Proceedings, *Electronic Publishing and the Information Superhighway*, Dartmouth Institute for Advanced Graduate Studies (DAGS), Boston, MA, May 30 - June 3, 1995.

[Davis 95] Dave Davis, ddavis@copyright.com, "A New Copyright Resource on the WWW from CCC," posted to *Public-Access Computer Systems Forum*, May 3, 1995.

[December 94] John December, "Challenges for Web Information Providers," from *The World Wide Web Unleashed*, Sams Publishing, forthcoming.

[Denning 95] Peter J. Denning, Bernard Rous, "ACM Electronic Publishing Plan and Interim Copyright Policies," *Communications of the ACM*, V. 38, No. 4, April 1995, pp. 97-109.

[Dozier 95] Linda T. Dozier, *NaviPress and NaviServer, A Client-Server Publishing System for the World-Wide Web*, NaviSoft, March 1995.

[DTIC 95] *Defense Technical Information Center*, at URL: http://www.dtic.dla.mil/gils/dtic-gils/DTIC.html, Last Updated June, 1995.

[EIT 94] Link-Checking Tool, *verify_links*, at URL: http://wsk.eit.com/wsk/dist/doc/admin/webtest/verify_links.html, Cited September 26, 1994.

[Enzel 95] Matisse Enzel, *Glossary of Internet Terms*, Internet Literacy Consultants, at URL: http://www.matisse.net, Cited July, 1995.

[Firewall 94] *Firewall Frequently Asked Questions*, at URL: http://www.cis.ohio-state.edu/hypertext/faq/usenet/firewall-faq/faq.html, Cited October, 1994.

[Flynn 94] Peter Flynn, *How to Write HTML Files*, at URL: http://kcgl1.eng.ohio-state.edu/www/doc/htmldoc.html, Cited October, 1994.

[FV 95] *First Virtual Holdings Incorporated*, at URL: http://www.fv.com, Cited August, 1995.

[Gloss-Soler 79] Shirley A. Gloss-Soler, *The DACS Glossary, A Bibliography of Software Engineering Terms*, Data & Analysis Center for Software, GLOS-1, October 1979.

[Graham 94] Ian Graham, *HTML Documentation*, at URL: http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/intro.html, Cited October, 1994.

[Grobe 94] Michael Grobe, *HTML Quick Reference*, Academic Computing Services, The University of Kansas, September 25, 1994, at URL: http://www.cc.ukans.edu/lynx_help/HTML_Quick.html.

[Hefley 94] William E. Hefley, *Exploring Hypermedia Information Services for Disseminating Software Engineering Information*, Software Engineering Institute, CMU/SEI-94-TR-03, February 1994.

[Horn 94] Wm. Dennis Horn, *HTML Documents: A Mosaic Tutorial*, at URL: http://fire.clarkson.edu/doc/html/htut.html, Cited October, 1994.

[Hughes 94] Kevin Hughes, *getstats*, a server log analysis tool, at URL: http://www.eit.com/software/getstats/getstats.html, Cited September 26, 1994.

[Interleaf 94] *Cyberleaf Product Sheet*, Interleaf, Inc., at URL: http://www.ileaf.com, Cited November, 1994.

[ISO 92] ISO/IEC, IS 10744, *Hypermedia/Time-based Document Structuring Language (HyTime)*, August, 1992.

[Jakes 89] J. Michael Jakes and E. Robert Yoches, "Legal Protection for Computer Software," *Communications of the ACM*, V. 32, No. 2, February 1989, pp. 169-171.

[Koster 94a] Martijn Koster, *ALIWEB Description Specification*, at URL: http://web.nexor.co.uk/aliweb/doc/format.html, Cited October, 1994.

[Koster 94b] Martijn Koster, *ALIWEB Submission Guidelines*, at URL: http://web.nexor.co.uk/aliweb/doc/guidelines.html, Cited October, 1994.

[Leslie 94] Jacques Leslie, "Goodbye, Gutenberg" *Wired*, V 2.10, October 1994, pp. 68-71.

[Lewis 93] Clayton Lewis, and John Rieman, *Task Centered User Interface Design*, shareware book published privately, University of Colorado, Boulder, CO, 1993; referenced in "Improved Usability from a System Upgrade," by Scott S. Heiner, *CrossTalk*, Vol. 7, No. 11, November 1994, p. 23.

[Lyman 94] Molly Lyman, E-mail, iactask-l@asc.dtic.dla.mil, DTIC hub page at URL: http://www.dtic.dla.mil/iac/; IAC Directory at URL: http://www.dtic.dla.mil/iac/iacdir.html, Cited October 29, 1994.

[Mayer 92] Richard J. Mayer, Arthur A. Keen and M.S. Wells, *IDEF4 Object-Oriented Design Method Manual*, Knowledge Based Systems, Inc., 1992.

[McBryan 94] McBryan, *Mother-of-all BBS*, at URL: http://www.cs.colorado.edu/homes/mcbryan/public-Html/bb/summary.html, Cited October 11, 1994.

[Munnecke 94] Tom Munnecke, *The World-Wide Web and the Demise of the Clockwork Universe*, the Second International Conference on Mosaic and the World Wide Web, October, 1994, at URL: http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Overviews/munnecke/www94.html.

[McGuire 94] James G. McGuire, "EIT-Link-Verifier-Robot/0.2," Georgia Institute of Technology; posted to *comp.infosystems.www.providers* on 8/3/94; tool at URL: http://wsk.eit.com/wsk/dist/doc/admin/webtest/verify_links.html, Cited October 1994.

[NCSA 94] NCSA, *A Beginner's Guide to HTML*, NCSA/pubs@ncsa.uiuc.edu, at URL: http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html, Cited October 19, 1994.

[NCSA 94b] NCSA, *Leasing Space on a Server*, at URL: http://union.ncsa.uiuc.edu/www/leasing.html, Cited October, 1994.

[Netscape 95] *Netscape Tell Us What's New Page*, at URL: http://home.netscape.com/escapes/submit_new.html, Cited August, 1995.

[NPS 95] US National Park Service, *Visit the Parks*, at URL: http://www.nps.gov/parks.html, Cited August, 1995.

[O'Brien 95] Maureen O'Brien, "Authors Organize to Launch Electronic Rights Service," *Publishers Weekly*, June 5, 1995, p. 10.

[Odlyzko 94] Andrew M. Odlyzko, *Tragic Loss or Good Riddance? The Impending Demise of Traditional Scholarly Journals*, at URL: http://gopher.econ.lsa.umich.edu/EconInternet/Information_ goods.html, July 16, 1994.

[Patch 94] Kimberly Patch, "Mosaic 'Secures' the Internet: Client-Based Encryption on Tap," *PC Week*, Vol. 11, No. 36, September 12, 1994, p. 8.

[Pathfinder 95] Time-Warner, *Welcome to Pathfinder*, at URL: http//www.pathfinder.com, Cited August, 1995.

[Patrias 91] Karen Patrias, *National Library of Medicine Recommended Formats for Bibliographic Citation*, US Dept. of Health and Human Services, April 1991.

[Pitkow 94] James Pitkow, *HTML_ANALYZER-0.30 README*, at URL: http://www.gatech.edu/pitkow/html_analyzer/README.html, Cited October, 1994.

[Quittner 94] Joshua Quittner, "Billions Registered," *Wired*, V. 2.10, October 1994, pp. 50-56.

[Raggett 95] Dave Raggett, *HyperText Markup Language Specification Version 3.0*, Internet Draft, at URL: http://ietf.cnri.reston.va.us/internet-drafts/draft-ietf/html/specv3-00.txt, March 28, 1995.

[RL 95] Rome Laboratory, Griffiss AFB, NY at URL: http://www.rl.af.mil, Cited August, 1995.

[Rees 94] Gareth Rees, *Publishing on the World Wide Web*, at URL: http://www.cl.cam.ac.uk/users/gdr11/publish.html, Cited September, 1994.

[Rosenfeld 94] Louis Rosenfeld, *Clearinghouse for Subject-Oriented Internet Resource Guides*, University Library, and School of Information and Library Studies, University of Michigan, at URL: http://www.lib.umich.edu/chhome.html, Cited October, 1994.

[Rutgers 94] Rutgers University Network Services www-security team, at URL: http://www-ns.rutgers.edu/www-security/index.html, Cited October, 1994.

[Samuelson 94a] Pamela Samuelson, "Copyright's Fair Use Doctrine and Digital Data," *Communications of the ACM*, Vol. 37, No. 1, January 1994, pp. 21-27.

[Samuelson 94b] Pamela Samuelson, "The NII Intellectual Property Report," *Communications of the ACM*, Vol. 37, No. 12, December 1994, pp. 21-27.

[Schneider 94] Karen G. Schneider, E-mail to PACS-L listserver, Cited November 18, 1994.

[SG 95] *Silicon Graphics' Silicon Surf*, Silicon Graphics, Incorporated, at URL: http://www.sgi.com, Cited July, 1995.

[Shaw 90] Mary Shaw, "Prospects for an Engineering Discipline of Software," *IEEE Software*, November 1990, pp. 15-24.

[Smartt 94] Mary Beth Smartt, "Introducing Cyberleaf," *Interleaves, The Interleaf Magazine*, November 1994, pp. 1-3.

[Smith 74] Ronald L. Smith, "Management Data Collection and Reporting," *Structured Programming Series*, RADC-TR-74-300, Vol. IX, Rome Air Development Center, 1974.

[Sorensen 92] Reed Sorensen, "Terms in Transition - Documentation Terminology," *CrossTalk, The Defense Software Engineering Report*, Software Technology Support Center, February 1992, p. 17.

[Sorensen 94] Reed Sorensen, "Documentation Management Awareness is Increasing," *CrossTalk, The Defense Software Engineering Report*, Software Technology Support Center, February 1994, pp. 22-24.

[SOW 91] *Statement of Work for Data and Analysis Center for Software (DACS)*, Rome Laboratory, Griffiss Air Force Base, PR No. C-2-2627, November 21, 1991.

[Stanford 95] Stanford University, *Yahoo - A Guide to WWW*, at URL: http://www.yahoo.com, Cited August, 1995.

[Stix 94] Gary Stix, "The Speed of Write, Trends in Scientific Communication," *Scientific American*, December 1994, p. 109.

[TANSU 94] Web Project, Telecom Australia - Network Systems, at URL: http://www.telstra.com.au/Info/security.html, Cited October, 1994.

[Tilton 93] James Tilton, *Composing Good HTML*, at URL: http://www.cs.cmu.edu/~tilt/cgh, Cited October, 1994.

[Torkington 93] Nathan Torkington, *A Primer on WWW Servers*, at URL: http://www.vuw.ac.nz/who/Nathan.Torkington/ideas/www-server.html, Cited October, 1994.

[Torkington 94] Nathan Torkington, *An Introduction to HTML*, at URL: http://www.vuw.ac.nz/non-local/gnat/www-html.html, Cited October, 1994.

[Trefzger 95] William Trefzger, *DTIC WWW Server Standards and Guidelines*, at URL: http://www.dtic.dla.mil/staff/trefzger/standards.html, April 24, 1995.

[Troetschel 95] Jay S. Troetschel, "My 2¢ Hits and Registration," *Advertising Age Marketing Forum*, at URL: http://www.adage.com/iaarchive/marketing.html, May 17, 1995.

[Unipress 94] Unipress Worldwide, *Weblint v1.010*, available at URL: http://www.khoros.unm.edu /staff/nelib/weblint.html, Updated November 21, 1994.

[Wallich 95] Paul Wallich, "The Chilling Wind Of Copyright Law?" *Scientific American*, February 1995, p. 30.

[Wiggins 94] Rich Wiggins, <WIGGINS@msu.edu>, "Obtaining Permission for Links," posted to *Library Gopher List* <GO4LIB-L@ucsbvm.ucsb.edu>, December 6, 1994.

[Wilder 95] Clinton Wilder and Jason Levitt, "Cure or Curse?" *Informationweek*, April 3, 1995, pp. 14-16.

[Wolf 94] Gary Wolf, "The [Second Phase of the] Revolution has Begun," *Wired*, V. 2.10, October 1994, pp. 116-121.

[Wong 94] Meng Weng Wong, *What Every Information Provider Should Know*, at URL: http://www.seas .upenn.edu/~mengwong/meng.html, April 23, 1994.

B.2 Sorted Bibliography

Categories
Internet Information

Net History

Net Culture

Other Internet Tools

Electronic Publishing Literature

Network Information

Addressing

Software Engineering Information

Human-Computer Interaction/User Interfaces

Design Literature

Design Examples

Personal Home Pages

HTML Guidelines

Forms Information

Graphics Information

Technical Details for Multimedia

Tools Lists

Services Information

Server Installation and Administration

Indexes and Search Systems

Intellectual Property Information

Security

Commerce

Standards/Guidelines

Data & Analysis Center for Software

Internet Information

Michael Strangelove, *Strangelove Internet Enterprises*, at URL: http://www.phoenix.ca/sie/

Leonard Klienrock, et al., *Realizing the Information Future - The Internet and Beyond*, Washington DC: National Academy Press, 1994.

Internet Newsgroup at URL: comp.infosystems.www, Cited October 20, 1994.

Newsgroup FAQ (Frequently Asked Questions) archive URL: ftp://rtfm.mit.edu

Yahoo - Computers and the Internet, at URL: http://www.yahoo.com/Computers_and_Internet/Internet/

[Enzel 95] Matisse Enzel, *Glossary of Internet Terms*, Internet Literacy Consultants, at URL: http://www.matisse.net, Cited July, 1995.

Wired Magazine, *Welcome To HotWired!* at URL: http://www.hotwired.com/

Net History

Charles Piller, "Dreamnet," *MacWorld*, October 1994, pp 96-105.

John Seabrook, "E-mail From Bill," *The New York Times*, January 10, 1994.

Net Culture

Aldea Communications, Inc. *Blue Netpages-Internet Survival Guide*, at URL: http://www.aldea.com /bluepages/blue.html

Charm Net Inc. *Charm Net Learning*, at URL: http://www.charm.net/learning.html

[Quittner 94] Joshua Quittner, "Billions Registered," *Wired*, V. 2.10, October 1994, pp. 50-56.

"Jargon Watch," *Wired*, V. 2.09, September 1994, p. 34.

Other Internet Tools

Cricket Liu, et al., *Managing Internet Information Services, First Edition*, Sebastopol, CA: O'Reilly & Associates, Inc., December 1994.

WAIS

freeWAIS, at URL: ftp://ftp.cnidr.org/pub/NIDR.tools/freewais

Telnet

NCSA Telnet, at URL: ftp://ftp.ncsa.uiuc.edu/Telnet/Mac/

Telnet, at URL: ftp://gatekeeper.dec.com/pub/micro/msdos/win3/winsock/

Electronic Publishing Literature

[DAGS 95] Proceedings, *Electronic Publishing and the Information Superhighway*, Dartmouth Institute for Advanced Graduate Studies (DAGS), Boston, MA, May 30 - June 3, 1995.

[December 94] John December, *Challenges for Web Information Providers*, at URL: http://www.rpi.edu/~decemj/cmc/mag/1994/oct/webip.html

[Leslie 94] Jacques Leslie, "Goodbye, Gutenberg" *Wired*, V 2.10, October 1994, pp. 68-71.

Jerome McGann, *The Rationale of Hypertext*, at URL: http://jefferson.village.virginia.edu/public/jjm2f /rationale.html

Samuel A. Rebelsky, *Electronic Publishing and the Electronic Superhighway*, at URL: http://www.cs.dartmouth.edu/~samr/dags95.html

Klaus Sullow, Rainier Page, "Hypermedia Browsing and the Online-Publishing Process," in Proceedings, *Electronic Publishing and the Information Superhighway*, Dartmouth Institute for Advanced Graduate Studies (DAGS), June 1995.

Network Information

[Berners-Lee 94a] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, "The World-Wide Web," *Communications of the ACM*, V. 37, No. 8, August 1994, pp. 76-82.

Addressing

T. Berners-Lee, *Universal Rescource Identifiers in WWW*, at URL: http://www.w3.org/hypertext/WWW/Adressing/URL/uri-spec.ps

Software Engineering Information

[Agresti 86] William W. Agresti, *New Paradigms for Software Development*, New York: IEEE Computer Society Press, 1986.

[Booch 86] Grady Booch, *Software Engineering with Ada, Second Edition*, Menlo Park, CA: Benjamin/Cummings Publishing Company, Inc., 1986.

Jacques Cohen, James Feinberg, *Communications of the ACM*, at URL: http://www.acm.org/pubs/cacm/

[Gloss-Soler 79] Shirley A. Gloss-Soler, *The DACS Glossary, A Bibliography of Software Engineering Terms*, Data & Analysis Center for Software, GLOS-1, October 1979.

Elaine Fedchak and Robert Vienneau, *A History of Software Measurement at Rome Laboratory*, Kaman Sciences Corporation, July 4, 1994.

Ian Parberry, *The Laboratory for Recreational Computing*, at URL: http://hercule.csci.unt.edu/larc

Roger Pressman, *Software Engineering: A Practicioner's Approach, Second Edition*, McGraw-Hill, 1987.

Mike Sendall, *Introduction to the Software Engineering Glossary*, at URL: http://dxsting.cern.ch/sting/glossary-intro.html

[Shaw 90] Mary Shaw, "Prospects for an Engineering Discipline of Software," *IEEE Software*, November 1990, pp. 15-24.

[Smith 74] Ronald L. Smith, "Management Data Collection and Reporting," *Structured Programming Series*, RADC-TR-74-300, Vol. IX, Rome Air Development Center, 1974.

Human-Computer Interaction/User Interfaces

Hans de Graff, *HCI Index: Table of Contents*, at URL: http://is.twi.tudelft.nl/hci/

[Lewis 93] Clayton Lewis, and John Rieman, *Task Centered User Interface Design*, shareware book published privately, University of Colorado, Boulder, CO, 1993; referenced in "Improved Usability from a System Upgrade," by Scott S. Heiner, *CrossTalk*, Vol. 7, No. 11, November 1994, p. 23.

Petre and Price, *Why Computer Interfaces Are Not Like Paintings: The User As A Deliberate Reader*, at URL: http://kunpu7.psych.kun.nl/pub/cogw/Petre-notPainting-EWHCI92.ps

Arkady Pogostkin and Svetlana Stenchikova, *Human Computer Interaction Laboratory at the University of Maryland*, at URL: http://www.cs.umd.edu/projects/hcil/

Design Literature

V. Balasubramanian, Bang Min Ma, and Joonhee Yoo, "A Systematic Approach to Designing a WWW Application," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 47-48.

[Balasubramanian 94] V. Balasubramanian, *State of the Art Review on Hypermedia Issues and Applications*, Graduate School of Management, Rutgers University, Newark, New Jersey, March 1994, at URL: http://www.isg.sfu.ca/~duchier/misc/hypertext_review, Cited August 1995.

Andreas Dieberger and Jay D. Bolter, "On the Design of Hyper 'Spaces'," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 98.

Franca Garzotto, Luca Mainetti, and Paolo Paolini, "Hypermedia Design, Analysis, and Evaluation Issues," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 74-86.

Paul Heckel, *The Elements of Friendly Software Design*, San Francisco: Sybex, Inc., 1991.

[Hefley 94] William E. Hefley, *Exploring Hypermedia Information Services for Disseminating Software Engineering Information*, Software Engineering Institute, CMU/SEI-94-TR-03, February 1994.

Tomás Isakowitz, Edward A. Stohr, and P. Balasubrumanian, "RMM: A Methodology for structured Hypermedia Design," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 34-44.

[Mayer 92] Richard J. Mayer, Arthur A. Keen and M.S. Wells, *IDEF4 Object-Oriented Design Method Manual*, Knowledge Based Systems, Inc., 1992.

[Munnecke 94] Tom Munnecke, *The World-Wide Web and the Demise of the Clockwork Universe*, Presented at the Second International Conference on Mosaic and the World Wide Web, October, 1994, at URL: http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Overviews/munnecke/www94.html.

Daniel Schwabe and Gustavo Rossi, "The Object Oriented Hypermedia Design Model," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 45-46.

[Sorensen 92] Reed Sorensen, "Terms in Transition - Documentation Terminology," *CrossTalk*, Issue 31, February 1992, p. 17.

Manfred Thüring, Jörg Hannemann, and Jörg M. Haake, "Hypermedia and Cognition Designing for Comprehension," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 57-66.

Design Examples

Paul Kahn, "Visual Cues For Local and Global Coherence in the WWW," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 67-69.

[Lyman 94] Molly Lyman, E-mail, iactask-l@asc.dtic.dla.mil, *DTIC hub page* at URL: http://www.dtic.dla.mil/iac/; IAC Directory at URL: http://www.dtic.dla.mil/iac/iadir.html, Cited October 29, 1994.

The Internet Movie Database at Mississippi US, at URL: http://www.msstate.edu/Movies/, Cited August, 1995.

NCSA Mosaic Demo Document, at URL: http://www.ncsa.uiuc.edu/demoweb/demo.html, Last Updated April 27, 1995.

[NPS 95] US National Park Service, *Visit the Parks*, at URL: http://www.nps.gov/parks.html, Cited August, 1995.

[Pathfinder 95] Time-Warner, *Welcome to Pathfinder*, at URL: http//www.pathfinder.com, Cited August, 1995.

Brandon Plewe, *The Best of WWW Contest*, at URL: http://wings.buffalo.edu.contest, Cited August, 1995.

Silicon Graphics Inc., *Silicon Graphics' Silicon Surf*, at URL: http://www.sgi.com.

[Stanford 95] Stanford University, *Yahoo - A Guide to WWW*, at URL: http://www.yahoo.com/, Cited August, 1995.

World-Wide Web Information, at URL: http://www.sei.cmu.edu/demos.html, Cited August, 1995.

The World-Wide Web Virtual Library: Subject Catalogue, at URL: http://www.w3.org/hypertext /DataSources/bySubject/Overview.html, Cited August, 1995.

Personal Home Pages

Douglas Cootey, *Doug's Dementia*, at URL: http://www.itsnet.com/home/bug/absent.html

Harry Plantinga, *Information about Harry Plantinga*, at URL: http://www.cs.pitt.edu/~planting/

Rocket897@aol.com, *Rocket J. Squirrel*, at URL: http://www.cygnus.com/misc/kfjc/rocketweb /rocketj4.html

[Wong 94] Meng Weng Wong, *What Every Information Provider Should Know*, at URL: http://www.seas.upenn.edu/~mengwong/meng.html, April 23, 1994.

HTML Guidelines

[Berners-Lee 94c] Tim Berners-Lee, *Style Guide for Online Hypertext*, at URL: http://www.w3.org /hypertext/WWW/Tools/Overview.html, Cited October 24, 1994; 1642 bytes.

[Connolly 94] Daniel W. Connolly, *HyperText Markup Language (HTML): Working and Background Materials*, at URL: http://www.w3.org/hypertext/WWW/MarkUp/, Cited November 3, 1994.

[Flynn 94] Peter Flynn, *How to Write HTML Files*, at URL: http://kcgl1.eng.ohio-state.edu/www/doc/htmldoc.html, Cited October, 1994.

[Graham 94] Ian Graham, *HTML Documentation*, at URL: http://www.utirc.utoronto.ca/HTMLdocs /NewHTML/intro.html, Cited October, 1994.

[Grobe 94] Michael Grobe, *HTML Quick Reference*, Academic Computing Services, The University of Kansas, September 25, 1994, at URL: http://www.cc.ukans.edu/lynx_help/HTML_Quick.html.

[Horn 94] Wm. Dennis Horn, *HTML Documents: A Mosaic Tutorial*, at URL: http://fire.clarkson.edu /doc/html/htut.html, Cited October, 1994.

Laura Lemay, *Teach Yourself Web Publishing With HTML in a Week*, at URL: http://slack.lne.com /lemay/theBook/

[NCSA 94] NCSA, *A Beginner's Guide to HTML*, NCSA/pubs@ncsa.uiuc.edu, at URL: http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html, Cited October 19, 1994.

Pierre Omidyar, *Pierre's Web Tips*, at URL: http://www.best.com/~pierre/web-authoring.html

[Rees 94] Gareth Rees, *Publishing on the World Wide Web*, at URL: http://www.cl.cam.ac.uk/users/gdr11 /publish.html, Cited September, 1994.

[Tilton 93] James Tilton, *Composing Good HTML*, at URL: http://www.cs.cmu.edu/~tilt/cgh, Cited October, 1994.

[Torkington 94] Nathan Torkington, *An Introduction to HTML*, at URL: http://www.vuw.ac.nz/non-local/gnat/www-html.html, Cited October, 1994.

Syed S. Towheed, *Candy Style* at URL: http://coney.gsfc.nasa.gov/www/sswg/candy_style.html

Forms Information

[Boutell 95] Thomas Boutell, *World Wide Web Frequently Asked Questions*, at URL: http://sunsite.unc.edu/boutell/faq/

Graphics Information

[Boutell 95] Thomas Boutell, *World Wide Web Frequently Asked Questions*, at URL: http://sunsite.unc.edu/boutell/faq/

Cricket Liu, et al., *Managing Internet Information Services, First Edition*, Sebastopol, CA: O'Reilly & Associates, Inc., December 1994.

Technical Details for Multimedia

Cricket Liu, et al., *Managing Internet Information Services, First Edition*, Sebastopol, CA: O'Reilly & Associates, Inc., December 1994.

National Technical Institute for the Deaf, *NDIT High Technology*, at URL: http://htc.rit.edu/

Compression

Compression FAQ, at URL: http://www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/top.html

Stuart Inglis, *Compression Pointers*, at URL: http://130.217.240.12/~singlis/compression-pointers.html

PKWARE® WWW Site Directory, at URL: http://www.pkware.com/index.html

Bandwidth/Communications Channels

Michael Perloff and Kurt Reiss, "Improvements to TCP Performance in High-Speed ATM Networks," *Communications of the ACM*, Vol. 38, No. 2, February 1995, pp. 90-100.

[Stix 94] Gary Stix, "The Speed of Write, Trends in Scientific Communication," *Scientific American*, December 1994, p. 109.

Ronald J. Vetter, "ATM Concepts, Architectures, and Protocols," *Communications of the ACM*, Vol. 38, No. 2, February 1995, pp. 39-44.

Tools Lists

Tools For WWW Providers, at URL: http://www.w3.org/hypertext/WWW/Tools/Overview.html

The SingNet WWW Team, *WWW & HTML Developer's JumpStation*, Version 2.0, at URL: http://oneworld.wa.com/htmldev/devpage/dev-page.html, Updated October 21, 1994.

Editors

[Berners-Lee 94b] Tim Berners-Lee, *W3 and HTMLTOOLS*, at URL: http://www.w3.org/hypertext/WWW/Tools/Overview.html, Cited October 21, 1994.

Converters

Nikos Drakos, *The LaTeX2HTML Translator*, Version 0.6.2, University of Leeds, at URL: http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/node6.html, August 27, 1994.

HoTMetaL, First Edition, SoftQuad, Inc., Toronto, CA, June 1994.

Kevin Hughes, *Hypermail,* Version 1.02, Enterprise Integration Technologies, at URL: http://www.eit.com/software/hypermail/hypermail.html.

rtftohtml for UNIX, Version 2.7, at URL: http://ftp.cray.com/src/WWWstuff/RTF/rtftohtml_overview.html

Jill Swift, *ANT_HTML.DOT, and ANT_PLUS.DOT,* Word for Windows 6.0 to HTML Conversion Utilities, October 28, 1994.

Kenneth Wong and Anton Lam, *CU_HTML.DOT,* Version 1.0, The Chinese University of Hong Kong, at URL: ftp://ftp.cuhk.hk/pub/pc/windows/winword/cu_html.zip, April 16, 1994.

Environments

[Dozier 95] Linda T. Dozier, *NaviPress and NaviServer, A Client-Server Publishing System for the World-Wide Web,* NaviSoft, March 1995.

[Interleaf 94] *Cyberleaf Product Sheet,* Interleaf, Inc., at URL: http://www.ileaf.com, Cited November, 1994.

MacWeb, at URL: ftp://ftp.einet.net/einet/mac/macweb/

Jocylene Nanard and Marc Nanard, "Hypertext Design Environments and Hypertext Design," *Communications of the ACM,* V. 38, No. 8, August 1995, pp. 49-56.

Navisoft InternetWorks, at URL: http://www.booklink.com/

[Smartt 94] Mary Beth Smartt, "Introducing Cyberleaf," *Interleaves, The Interleaf Magazine,* November 1994, pp. 1-3.

Norbert A. Streitz, "Designing Hypermedia: A Collaborative Activity," *Communications of the ACM,* V. 38, No. 8, August 1995, pp. 70-71.

[Wolf 94] Gary Wolf, "The [Second Phase of the] Revolution has Begun," *Wired,* V. 2.10, October 1994, pp. 116-121.

Browsers, by Platform:

Macintosh

NCSA Mosaic, at URL: http://www.ncsa.uiuc.edu/SDG/Software/MacMosaic/MacMosaic Home.html
Netscape Navigator, at URL: http://home.netscape.com/info/how-to-get-it.html

Microsoft Windows

NCSA Mosaic for Microsoft Windows Home Page, at URL: http://www.ncsa.uiuc.edu/SDG /Software/WinMosaic/HomePage.html
Netmanage WebSurfer, at URL: http://www.netmanage.com/cgi-bin/imagemap /inetcham?233,47
Quarterdeck Mosaic, at URL: http://www.qdeck.com/

NeXTStep

OmniWeb at URL: http://www.lighthouse.com/ProductInfo/OmniWeb.html
CERN NEXTSTEP browser, at URL: ftp://ftp.w3.org/pub/www/bin/next/
Netsurfer, at URL: http://software.thoughtport.com/Products/Netsurfer.html
SpiderWoman, at URL: http://sente.epfl.ch/~swoman/

OS/2

IBM WebExplorer, at URL: ftp://ftp.ibm.net/pub/WebExplorer/

X-Windows

HotJava, at URL: http://java.sun.com/
NCSA Mosaic, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Unix/binaries/
Netscape Navigator, at URL: ftp://ftp.mcom.com/netscape/unix/

Testers

[Churchyard 95] Henry Churchyard, *htmlchek - HTML Error Checker and Utilities*, Version 4.0, University of Texas at Austin, at URL: http://www.w3.org/hypertext/WWW/Tools/htmlchek.html, Updated January 23, 1995.

[EIT 94] Link-Checking Tool, *verify_links*, at URL: http://wsk.eit.com/wsk/dist/doc/admin/webtest/verify_links.html, Cited September 26, 1994.

Roy Fielding, *MOMspider -- Distribution Information*, at URL: http://www.ics.uci.edu/WebSoft/MOMspider, Cited August, 1995.

Scott Johnson, "Quality Control For HyperText Construction," *Communications of the ACM*, Vol. 38, No. 8, August 1995, pp. 87.

[McGuire 94] James G. McGuire, *EIT-Link-Verifier-Robot/0.2*, Georgia Institute of Technology; posted to *comp.infosystems.www.providers* on 8/3/94, Tool at URL: http://wsk.eit.com/wsk/dist/doc/admin/webtest/verify_links.html, Cited October 1994.

[Pitkow 94] James Pitkow, *HTML_ANALYZER-0.30 README*, at URL: http://www.gatech.edu/pitkow/html_analyzer/README.html, Cited October, 1994.

[Unipress 94] Unipress Worldwide, *Weblint v1.010*, available at URL: http://www.khoros.unm.edu/staff/nelib/weblint.html, Updated November 21, 1994.

Log Analyzers

[Hughes 94] Kevin Hughes, *getstats*, a server log analysis tool, at URL: http://www.eit.com/software/getstats/getstats.html, Cited September 26, 1994.

Daniel LaLiberte, *HTTPD Log Analyzers*, at URL: http://union.ncsa.uiuc.edu/HyperNews/get/www/log-analyzers.html

Yahoo - Computers and Internet:Internet:World Wide Web:HTTP:Servers:Log Analysis Tools, at URL: http://www.yahoo.com/Computers/World_Wide_Web/HTTP/Servers/Log_Analysis_Tools/

Viewers

[Adobe 94] *Adobe Acrobat*, at URL: http://www.adobe.com/Acrobat/Acrobat0.html, Cited October 28, 1994, 3190 bytes.

WWW Browsers and Applications, at URL: http://www.sdic.com/WWWStuff.html

Gianfranco Pocecai, *WWW Browsers*, at URL: http://www.dsi.unimi.it/Users/Students/gpoc/w3a/browsers.html

Macintosh

Helper Applications, at URL: http://www.ncsa.uiuc.edu/SDG/software/MacMosaic/helpers.html

Related Applications, at URL: http://www.ncsa.uiuc.edu/SDG/software/MacMosaic/related.html

Audio

SoundMachine, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

RealAudio Player, at URL: http://www.realaudio.com/

Decompression

StuffitExpander, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

Images

GIFConverter, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

GraphicConverter, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

JPEGView, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

PDF (Portable Document Format)

Acrobat Reader, at URL: http://www.adobe.com/Software.html#acrordr

Video

Sparkle, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

Fastplayer, at URL: ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/

SimplePlayer, at URL: http://www.mcad.edu/Guests/EricB/xplat.html#mov.qt

MS-DOS

Jih-Shin Ho, *DISPLAY Version 1.87,* at URL: ftp:// NCTUCCA.edu.tw/PC/graphics/disp

Photodex Corporation, *Graphics Display Sytem (GDS) Version 3.1e,* at URL: ftp://ftp.netcom.com/pub/ph/photodex

Microsoft Windows

Mosaic and External Viewers, at URL: http://www.ncsa.uiuc.edu/SDG/software/WinMosaic/viewers.htm

Audio

RealAudio Player, at URL: http://www.realaudio.com/

Images

LView,at URL: ftp://oak.oakland.edu/SimTel/win3/graphics/

WINGIF, at URL: ftp://ftp.spry.com/ms-windows/viewers/gif/apps/

WINJPEG, at URL: ftp://ftp.spry.com/ms-windows/viewers/jpeg/apps/

PDF (Portable Document Format)

Acrobat Reader, at URL: http://www.adobe.com/Software.html#acrordr

PostScript

Ghostscript, at URL: http://www.cs.wisc.edu/~ghost

Ghostview, at URL: http://www.cs.wisc.edu/~ghost

Video

GPlay, at URL: ftp://gatekeeper.dec.com/pub/micro/msdos/win3/desktop/ MPE AVI Video

mpegnt, at URL: ftp://wuarchive.wustl.edu/systems/ibmpc/win3/nt/

QuickTime for Windows, at URL: ftp://ftp.support.apple.com/pub/apple_sw_updates /US/DOS%20%26%20Windows/QuickTime%20For%20Windows/

NeXTStep

Images

OmniImageFilter, at URL: http://www.omnigroup.com/Software/OmniImageFilter/

Audio

GISO, at URL: ftp://ftp.cs.orst.edu/pub/next/binaries/sound/

Video

NEXTIME, at URL: http://www.next.com/Info/Products/Ordering.html
Movie, at URL: ftp://ftp.cs.orst.edu/pub/next/binaries/graphics/

OS/2

PostScript

PM GSview, at URL: http://www.cs.wisc.edu/~ghost

X-Windows

NCSA Mosaic FAQ: Other Mosaic/WWW Software, at URL: http://www.ncsa.uiuc.edu /SDG/XMosaic/faq.software.html

Audio

showaudio, at URL: ftp://unix1.andrew.cmu.edu/pub/AUIS/auis-6.3/overhead/mail /metamail/scripts

Images

xv, at URL: file://ftp.ncsa.uiuc.edu/Mosaic/Unix/viewers/

PostScript

Ghostscript, at URL: http://www.cs.wisc.edu/~ghost

Video

MPegplay, at URL: file://ftp.ncsa.uiuc.edu/Mosaic/Unix/viewers/

xanim, at URL: ftp://sunsite.unc.edu/pub/X11/contrib/applications/

Other

Sha Xin Wei *Metamedia DistributedDatabase,* at URL: http://lummi.stanford.edu/Media2/ASD/ASD_Homepage/Multimedia.html

Services Information

Publicize Your Home Page, at URL: http://www.inforamp.net/ntg/public.html

Yahoo-Business:Corporations:Internet Presence Providers:Consulting, at URL: http://www.yahoo.com/Business/Corporations/Internet_Presence_Providers/Consulting, Cited August, 1995.

Server Installation and Administration

Sam Kimery, "Going Public," *SunWorld,* May 1993, pp.103-104.

[NCSA 94b] NCSA, *Leasing Space on a Server,* at URL: http://union.ncsa.uiuc.edu/www/leasing.html, Cited October, 1994.

Jon Wiederspan, *WWW Developer Resources,* at URL: http://www.uwtc.washington.edu/computing/www/general.html

Indexes and Search Systems

[Koster 94a] Martijn Koster, *ALIWEB Description Specification,* at URL: http://web.nexor.co.uk/aliweb/doc/format.html, Cited October, 1994.

[Koster 94b] Martijn Koster, *ALIWEB Submission Guidelines,* at URL: http://web.nexor.co.uk/aliweb/doc/guidelines.html, Cited October, 1994.

Lycos Search Engine at URL: http://www.lycos.com.

Lycos Registration Form, at URL: http://www.lycos.com/lycos-register.html

[McBryan 94] McBryan, *Mother-of-all BBS,* at URL: http://www.cs.colorado.edu/homes/mcbryan/public-Html/bb/summary.html, Cited October 11, 1994.

[Rosenfeld 94] Louis Rosenfeld, *Clearinghouse for Subject-Oriented Internet Resource Guides,* University Library, and School of Information and Library Studies, University of Michigan, at URL: http://www.lib.umich.edu/chhome.html, Cited October, 1994.

WebCrawler Search Engine at URL: http://webcrawler.com.

WebCrawler Submission, at URL: http://webcrawler.com/WebCrawler/SubmitURLS.html

World Wide Web Worm at URL: http:// www.cs.colorado.edu/home/mcbryan/wwww.html

Intellectual Property Information

APA Style Manual, 4th Edition, contains information on citing material from a variety of electronic sources, including E-mail and FTP.

[Davis 95] Dave Davis, ddavis@copyright.com, "A New Copyright Resource on the WWW from CCC," posted to *Public-Access Computer Systems Forum,* May 3, 1995.

[Denning 95] Peter J. Denning and Bernard Rous, "ACM Electronic Publishing Plan and Interim Copyright Policies," *Communications of the ACM,* V. 38, No. 4, April 1995, pp. 97-109.

Fred Greguras, Michael R. Egger, Sandy J. Wong, *Multimedia Content and the Super Highway: Rapid Acceleration or Foot on the Brake?* at URL: ftp://ftp.eff.org/pub/CAF/law/multimedia-copyright

[Jakes 89] J. Michael Jakes and E. Robert Yoches, "Legal Protection for Computer Software," *Communications of the ACM,* V. 32, No. 2, February 1989, pp. 169-171.

Electronic Style : a Guide to Citing Electronic Information,Westport, Connecticut: Meckler, 1993. [ISBN: 0-88736-909-X]

[O'Brien 95] Maureen O'Brien, "Authors Organize to Launch Electronic Rights Service," *Publishers Weekly*, June 5, 1995, p. 10.

[Patrias 91] Karen Patrias, *National Library of Medicine Recommended Formats for Bibliographic Citation*, US Dept. of Health and Human Services, April 1991.

[Samuelson 94a] Pamela Samuelson, "Copyright's Fair Use Doctrine and Digital Data," *Communications of the ACM*, V. 37, No. 1, January 1994, pp. 21-27.

[Samuelson 94b] Pamela Samuelson, "The NII Intellectual Property Report," Pamela Sameulson,"Copyright and Digital Libraries," *Communications of the ACM*, V. 37, No. 12, December 1994, pp. 21-27.

Pamela Sameulson,"Copyright and Digital Libraries," *Communications of the ACM*, V. 38, No. 4, April 1995, pp. 15-21.

[Wallich 95] Paul Wallich, "The Chilling Wind Of Copyright Law?" *Scientific American*, February 1995, pp 30.

[Wiggins 94] Rich Wiggins, <WIGGINS@msu.edu>, "Obtaining Permission for Links," posted to *Library Gopher List* <GO4LIB-L@ucsbvm.ucsb.edu>, December 6, 1994.

Security

Jon A. Rochlis and Mark W. Eichin, "With Microscope and Tweezers: The Worm from MIT's Perspective," *Communications of the ACM*, V. 32, No. 6, June 1989, pp. 689-698.

Muffy Barkocy, *SATAN*, (Security Administrator's Tool for Analyzing Networks) at URL: http://www.fish.com/~zen/satan/satan.html

Daniel Stevenson, Nathan Hillery, and Greg Byrd, "Secure Communications In ATM Networks," *Communications of the ACM*, Vol. 38, No. 2, February 1995. pp. 45-53.

[Wilder 95] Clinton Wilder and Jason Levitt, "Cure or Curse?" *Informationweek*, April 3, 1995, pp. 14-16.

Encryption

[Patch 94] Kimberly Patch, "Mosaic 'Secures' the Internet: Client-Based Encryption on Tap," *PC Week*, Vol. 11, No. 36, September 12, 1994, p. 8.

Organizations

[CERT] Computer Emergency Response Team advisories are available via anonymous FTP from cert.org.

[Rutgers 94] Rutgers University Network Services www-security team, at URL: http://www-ns.rutgers.edu/www-security/index.html, Cited October, 1994.

[TANSU 94] Web Project, Telecom Australia - Network Systems, at URL: http://www.telstra.com.au/Info/security.html, Cited October, 1994.

Firewalls

[Cheswick 94] William R. Cheswick and Steven M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Reading, Mass: Addison-Wesley, 1994.

[Firewall 94] *Firewall Frequently Asked Questions*, at URL: http://www.cis.ohio-state.edu/hypertext/faq/usenet/firewalls-faq/faq.html, Cited October, 1994.

Commerce

[Artner 95] Bob Artner, "The Revenge of the Press Release!" *Internet Business Advantage*, Vol. 1, No. 3, July 1995.

"CARI on, my Webward Businesses," *Internet Business Advantage*, V. 1, No. 2, June 1995, p.14.

[FV 95] *First Virtual Holdings Incorporated*, at URL: http://www.fv.com, Cited August, 1995.

[Odlyzko 94] Andrew M. Odlyzko, *Tragic Loss or Good Riddance? The Impending Demise of Traditional Scholarly Journals*, at URL: http://gopher.econ.lsa.umich.edu/EconInternet/Information_goods.html, July 16, 1994.

[Troetschel 95] Jay S. Troetschel, "My 2¢ Hits and Registration," *Advertising Age Marketing Forum*, at URL: http://www.adage.com/iaarchive/marketing.html, May 17, 1995.

Standards/Guidelines

[Christian 94] Eliot Christian, *GILS*. available on the FedWorld electronic bulletin board [703-321-8020] or via FTP at info.er.usgs.gov in the directory /public/gils as gils.doc [Microsoft Word for Windows 6.0 format], gils.rtf [Rich Text Format], or gils.txt [ASCII text format], Cited November 3, 1994.

[Connolly 94] Daniel W. Connolly, *HyperText Markup Language (HTML): Working and Background Materials*, at URL: http://www.w3.org/hypertext/WWW/MarkUp/, Cited November 3, 1994.

GILS at URL: http://info.er.usgs.gov/public/gils/gilsdoc.html, Cited November 3, 1994.

[Trefzger 95] William Trefzger, *DTIC WWW Server Standards and Guidelines*, at URL: http://www.dtic.dla.mil/staff/trefzger/standards.html, April 24, 1995.

An Overview of Four HTML Converters:

rtftohtml for UNIX ver. 2.7

CU_HTML.DOT (ver. 1.0) for Word for Windows 6.0

LaTeX2HTML Translator (ver. 0.6.2)

Prepared by:

Joe Borgia and Brian Davies

Hypermail (ver. 1.02)

Prepared by:

Brian Davies

Introduction to this Document

The following are overviews of four publicly available HTML converters. The text describes the experiences we had with these converters. This section should not be taken as an endorsement for a specific application of any of these converters. Rather, the idea is to give the reader a starting point for exploring existing converter options. Information about obtaining the conversion tools discussed is located at the end of each overview.

C.1 rtftohtml for UNIX--Version 2.7

This tool will take documents which are in Rich Text Format (RTF) form and convert them into HTML documents. It converts RTF files with tables, graphics, equations, and text to HTML. The test cases used with this tool had no images included in the original RTF document, therefore no examination was made of either how the images would be converted into the resulting HTML file or their quality after conversion. The UNIX operating system version of this tool was examined for this overview. This converter is also available in a Macintosh version.

The process of using rtftohtml 2.7 is simple. The tool itself is easy to use. When you have the file you want converted, type the command:

rtftohtml <filename.rtf>

Before you type this command, you must have the file in the same directory as the *rtftohtml* tool. This is an important point because it will not convert your RTF file otherwise.

From there, rtftohtml produces up to five different types of HTML files per RTF file conversion. The number of files produced depends upon the types of entities encountered within the RTF file (e.g., graphics, footnotes, headers, etc.). The five types of files are:

1.*filename.html* (the actual converted RTF file)

2.*filename_ToC.html* (a table of contents document in HTML)

3.*filename_fn.html* (an HTML document of footnotes if any exist)

4.*filename<1..n>.gif* (for graphics files, if any exist)

5.*filename.err* (file where errors or warnings are put if any occur during the conversion)

During conversion, these files are created automatically. The links between each of these files are also generated automatically.

The following is a break-down of how these files may be produced and what they may contain:

1.*The Resulting HTML file:* This document contains the converted RTF file written in HTML code.

2.*Table of Contents Document:* If headers exist within the original RTF file, they are made into a table of contents file to link each individual section within the converted file. Each header becomes an entry in this document and is a hypertext link to the location of its corresponding text.

3.*Footnote Document:* If footnotes exist in the original RTF file, they are turned into links in the converted document. The text of the footnote is stored in an HTML footnote document. In the converted document, you click on the footnote number to link to that HTML footnote document.

4.*Graphics files:* Graphics within the original RTF document are numbered and given the extension *.gif.* The files have to be in *GIF* format because most browsers have trouble reading anything but GIF format graphics.

5.*Error Document:* This is the file in which errors or warning messages that may have been produced during the conversion are placed.

This conversion tool can be obtained from the following places:

FTP:

ftp.utica.kaman.com

directory: *pub/HTML-Converters/unix/rtftohtml2.7.tar.Z*

WWW:

URL: *http://ftp.cray.com/src/WWWstuff/RTF/rtftohtml_overview.html*

This tool was also found at various sites as indicated by an Archie search on the Internet.

C.2 CU_HTML.DOT 1.0 (Word for Windows 6.0 Template)

CU_HTML.DOT is a document template that can be used in Microsoft Word for Windows 6.0. It can also be used with Word 2.0 but, for the purposes of this test it was only used with Word 6.0. As a new template within Word, a new *HTML* item is added to the menu bar at the top of the screen. Also, six new buttons are added to the toolbar which provide a short-cut to the template's functions.

This template allows a user to create a document in Word in a What You See Is What You Get (WYSIWYG) manner and then convert it to an HTML file. Before someone uses this tool, it is suggested that they have a competent, working knowledge of HTML code. CU_HTML itself does not go a long way to convert a document. It converts documents on a very superficial scale. After the document is converted, there is a great chance that the resulting HTML file will have to be manually edited to the user's liking. Expect to edit HTML documents created with this tool to tweak to exact requirements.

Nonetheless, this tool will reduce the time it would normally take to produce an HTML document from just manual coding. However, there are tradeoffs. With this tool, there is no automatic logical linking of various sections of a document as in other tools. In the test document that was used, the various sections of the document had to be broken up and stored as separate files so links could be inserted to separate sections. A table of contents existed in this document which was used as the top-level HTML file to link all of the sub-sections together. With this in mind it is important to remember that there is a design element which needs to be considered before you use this tool on a document. Before you start converting plan out what you expect the final HTML document to look like. This will simplify the pre-conversion layout as well as the post-conversion editing process.

Overall, the tool is simple to use. The easiest way to use the tool is to open a document in Word, then copy and paste it into a new document with the HTML template. From there on, you can start preparing your document for conversion to HTML.

This tool does allow for this insertion of images into the resulting HTML document which were in the original Word document. The problem is that in order to include these images in the resulting HTML file, they have to be in *.gif* format. If you have images in your Word document, there is a good chance that they are not in this format. These will have to be converted to the right format and saved as .gif files *before* conversion. The images in the test case were created within the Word drawing application. To convert them, the image was selected, then copied and pasted into *Paintbrush*, another graphic application. From this application the images were saved as *.pcx* files, which is a *Paintbrush* file format. Following this, the images were opened in a graphics conversion tool, *Graphics Workshop*. This tool allows the conversion of a graphic stored in one format to another format; in this case .pcx to .gif format. Note that if you do not convert your images to .gif format, they will not be included in your resulting HTML file after conversion. To add this image to the HTML document within Word, place the cursor where the image should appear in the HTML document and click on the *Image Button* on the toolbar. A pop up window will appear asking where the image is located. When you indicate where it is, the image source code is embedded in the Word document.

To create links to another HTML file, highlight the word you want to become the link (the "hotword"), click on the *Link Button* to pop up the file menu, and tell it where to find the file to link to. The same holds true for URLs. You can associate a URL with a hotword just as easily. To do this, click on the *URL Button* from the toolbar to pop up a window which prompts you for the URL that you want to link to.

It is important to note that all of these actions of linking image sources, other HTML files, or URLs must be done *before* the main conversion is done. Otherwise, the images and links that you want will not be included in the converted file. To convert the file, you click on the *Write HTML* button on the toolbar.

Data & Analysis Center for Software

The tool will then cycle through the Word document, generating the needed HTML code. When it is finished, the tool saves the document to a *.htm* file.

Ultimately this tool provides a template which will accomplish most of the HTML conversion work while allowing you to edit using a familiar interface. You will still have to do a lot of manual editing to get the final HTML document to look like you want it to look, but most users will probably find that editing easier if they are using a simple GUI interface like MS Word. CU_HTML can help cut down on development time and provide a friendly interface, but users need to understand how HTML works to use it because CU_HTML cannot do the whole job by itself.

This tool is available at the following site:

FTP:

ftp.cuhk.hk

Directory: */pub/pc/windows/winword/cu_html.zip*

It was also found at numerous other FTP sites through the results of an Archie search.

C.3 LaTeX2HTML Translator (Version 0.6.2)

This converter allows documents created with LaTeX to be converted into HTML documents. It is supposed to be able to convert LaTeX documents containing graphics and equations as well as text created within LaTeX. However, because the test case that was used for this conversion tool contained no images or equations, the effectiveness of this feature of LaTeX2HTML was not tested.

To run this tool, a general knowledge of UNIX and the general structure of LaTeX documents is recommended. Also, you should have a general knowledge of dealing with UNIX configuration files. Without a solid working knowledge of UNIX and how configuration files work, installation of this tool may prove to be difficult. LaTeX2HTML is a PERL application. To run it properly, your system will need to be running PERL version 4 at patch level 36.

This tool will take a LaTeX document with all of its component files and separate them into interlinked HTML files. The top level HTML document created is in the form of a table of contents which contains links to the input files which make up the main "master" LaTeX document. To execute the conversion, the main "master" LaTeX document and the input LaTeX files that are passed to that main document all need to be in the same directory. If LaTeX2HTML is not installed with LaTeX on your system, or linked for use, copy the "master" and input files into the directory with the LaTeX2HTML converter files. This is the same directory which contains the LaTeX2HTML converter executable. The only file that needs to be converted is the master LaTeX file. After that, all of the separate LaTeX files that make up the master LaTeX document are automatically converted.

In the test case, the master LaTeX file was called *template.tex*. To translate this file to HTML, at the command line prompt type:

   latex2html <filename.tex>

The test case command line looked like this:

   latex2html template.tex

There are various command line options for customizing a translated document during the conversion process which are outlined in a comprehensive manual included with the tool.

When the translation is completed, a directory is created by the tool which bears the name of the converted document. In the test case for example, a directory called *template* was created. This directory contains all of the resulting HTML files. These include the top level HTML document, which contains the table of contents and links to the rest of the HTML files also located in that directory. LaTeX2HTML separates the sub-sections of the original document into separate HTML files with links to those files in the top level document. This tool also allows for hypertext links to any existing footnotes and references included in the document.

Navigation tools for each document are included with the LaTeX2HTML tool. These are in the form of three buttons which are appended to the top of each resulting HTML document. The buttons are labeled *Next, Up,* and *Previous.* This allows for sequential reading of the document. At the bottom of each page, the tool also automatically appends the identity of the person who made the translation, when, and what time they made it. This can be inconvenient, especially if the translation was performed by someone other than the document's author. However, even if this is an undesirable "feature" of this tool, the resulting HTML document can be easily edited and the translator's name removed, or replaced with the author's name, from every HTML file.

Examples of LaTeX documents converted with LaTeX2HTML can be found at:

http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/node6.html

This tool can be obtained at the following sites:

WWW:

http://cbl.leeds.ac.uk/nikos/tex2html/latex2html/latex2html.tar

WWW (FTP):

ftp://ftp.tex.ac.uk/pub/archive/support/latex2html

C.4 Hypermail 1.02 from Enterprise Integration Technologies (EIT)
C version by Kevin Hughes

(Hypermail is free for non-commercial uses, but all users should read the license agreement which accompanies the software.)

This tool will convert UNIX mailbox format messages into a set of cross-referenced HTML documents. Archives created by Hypermail allow easy searching of topics, and following topic threads. The tool is most applicable for archiving E-mail lists, but could also be used for archiving postings from a USENET newsgroup.

Hypermail is easy to use and configure. One caveat, however. The tool is only available for UNIX systems, and even then can only interpret messages saved into plain vanilla UNIX-style mailbox format. Users of other popular UNIX mail packages, such as Rand MH, or the University of Washington's Pine, will have to deal with the less intuitive 'mail' interface to set up the mailbox file, commonly called 'mbox', that hypermail uses as a default input file.

Hypermail is easy to set up and can be retrieved as either a pre-compiled binary for most systems, or as source code to be compiled locally. I retrieved the source code and the pre-compiled binary for SunOS 4.1.3 machines. Hypermail compiled and installed without difficulty. Following the initial setup and implementation, Hypermail can be configured to automatically update archives.

Invoking Hypermail is even easier than installing it. The syntax to run it using a mailbox format file other than mbox (which Hypermail uses as a default) is:

hypermail -m <mailbox_formatted_file> -d <archive_directory> -l <archive_label>

Hypermail can also be configured to use standard input instead of a specific mailbox file. To do so replace the -m <mailbox_formatted_file> with a -i. Using this option will also you to setup Hypermail up to automatically archive new messages.

Running Hypermail produces a number of HTML files. Each individual message, in the mailbox file or from standard input, is output by Hypermail sequentially as *0000.html, 0001.html, 0002.html, 0003...* and so on. Along with the actual files to be references, the program generates the accompanying index files and links between them. The following, taken from the UNIX "man pages" for Hypermail, describes each of the index files created.

date.html
> The index of articles sorted by the date they were received by the mail daemon.

thread.html
> The index of articles sorted by thread first, then the date they were received.

subject.html
> The index of articles sorted by subject. Any Re: pre-fixes in front of subjects will have been stripped out.

author.html
> The index of articles sorted by the first word of the author's name. If the author's name can't be determined, the author's E-mail address will be substituted.

In addition to these, the tool generates the file *index.html* which is the default index users can use to search the archive.

Examples of archives created with Hypermail can be examined at:

WWW:

>*http://www.eit.com/mailinglists/lists.html*

The most up-to-date documentation about Hypermail (and the tool itself) can be retrieved from EIT directly.

>**WWW:**

>*http://www.eit.com/software/hypermail/hypermail.html*

You can also ftp Hypermail directly from EIT.

FTP:

>*ftp ftp.eit.com*

>directory: *pub/web.software/hypermail*

APPENDIX D SAMPLE GUIDELINES

This appendix contains two documents from the Defense Technical Information Center that are illustrative of handbook concepts. Section D.1 is a copy of a guidelines document developed at DTIC to provide agency-specific guidance for authoring HTML documents that are included in the DTIC WWW Information Kiosk [Trefzger 95]. Section D.2 is a copy of the DTIC Web page that was created in response to the Government Information Locator Service (GILS) definition of required core elements [DTIC 95].

D.1 DTIC WWW Server Standards and Guidelines
Final DRAFT (4/24/95)

Note: This is the final draft, which will be made available as the first version if there are no further comments.

Please address any questions or comments about this document to Bill Trefzger (wtrefzge@dgis.dtic.dla.mil).

Categories:
Background
I. Content
A. DTIC/DoD-produced content
B. External links
II. Navigation/Organization
III. Style
A. Interface
B. Technical
IV. DTIC-specific issues
V. Additional recommendations and points to ponder

Background

This document lists the specific standards and general guidelines by which DTIC will make information available on the World Wide Web (WWW). It is expected that there will be a wide variation in the missions, goals and contents of the various WWW-based projects at DTIC. This document will not attempt to provide answers to every possible situation that may arise as these projects are implemented.

It is, however, the goal of this document to provide enough standardization in DTIC WWW projects to ensure a high-quality presentation and consistency for users. It is also the goal of this document to help those implementing WWW services at DTIC to identify the issues which should be addressed as a project is developed and made available to users.

It is expected that, equipped with this document, plus an understanding of DTIC's mission, and the particulars of a given project or service, that project implementors at DTIC will develop the best possible services, and that these services will have enough common elements to identify them as DTIC-produced.

This document does not serve as a training document for WWW users or providers. It also does not serve as complete documentation of the procedures for project leaders to follow for publishing information on DTIC's web server; although certain procedural issues are addressed.

This list will be revised as necessary. Latest revision: 4/24/95

Legend
> #= standard, compliance required
> *= guideline, compliance recommended

Note: The requirement of particular projects may take precedence over one or more the standards below. DTIC project leaders must ensure the compliance of their services with the following in the absence of a contradictory requirement. More details are in the DTIC-Specific Issues.

I. Content

A. DTIC/DoD-produced content

Statement of Purpose.
> The content of all pages on the DTIC http server shall be related to the function and mission of DTIC. A specific statement describing the purpose and content of a collection will be included in (or linked to/from) the Project Home Pages.
>
> This statement can and should be considered the "collection policy" for the project. It can be used as a measure of whether a document or link should be included.

Approvals.
> The approval for the content of a given document or documents is to be made at the directorate level. This includes personal home pages.
>
> New or substantially changed projects will be announced (prior to their public release) in house to other DTIC directorates (see DTIC-specific issues).
>
> * An "Approved for public release" statement may be appropriate.

DoD Policy Compliance: Clearance of electronic information
> All information on the DTIC http server will be in compliance with the latest policy directions from OSD. See the memo from Deputy Secretary of Defense John Deutch and DoD Directive on release of information to the public.

DoD Policy Compliance: Government Information Locator Service (GILS) record creation
> Information sources on the DTIC http server will (when appropriate) have GILS records created for them, so that the source may be identified using the GILS services provide by DoD an others. DoD-wide guidance on GILS is expected in July 1995; initial instructions are available before then.

Responsibility.
> Every document (or document collection) will have an E-mail address, or a link to an E-mail form, which can be used to contact a responsible party regarding the content. This will not necessarily be a DTIC staffer.

Timeliness.

Documents must be kept up to date. Out-of-date information must be removed immediately.

Compliance.

Project leaders and staff working on their specific projects are directly responsible for the content of the documents in their collections, and ensuring that these documents are consistent with the statement of purpose and DTIC's mission.

Directorate Home Pages.

WWW documents describing the organization and activities of different directorates at DTIC are mandated. Directorate Home Pages will link to the DTIC Home Page, and provide links to the Personal Home Pages in that directorate (and of course, other relevant documents on the DTIC server).

Personal Home Pages.

Personal Home Pages are permitted if they relate to and support the functions the individual performs at DTIC. All information and external links on a Personal Home Page should support this purpose. They should be considered Project Home Pages, in that there will be a documented purpose. A link to the standard disclaimer http://www.dtic.dla.mil/dtiw/disclaimer.html must be included near the top of the page. Individuals are responsible for making sure the content of their personal home pages and related documents are appropriate and approved by their directorate management. A Personal Home Page should not be considered as an "electronic office" or an "electronic desktop". Things that might be harmlessly posted on an office wall are not necessarily appropriate to the purpose of a Personal Home Page.

B. External Links

External Link Defined.

An external link is a link to a document that is not on a DTIC server.

Approval.

The decision to include a link to an external source should be based on the statement describing the purpose of a document or project.

Context.

The context in which a link is made to external sources must be considered. It is important not to give the impression, for example, that DTIC is endorsing a commercial product. It is also important not to give the impression that DTIC is linking to frivolous (or worse) information sources. Statements about why the link(s) are provided may be appropriate or necessary to provide the context to the user.

Coordination.

The Defense Technical Information Web (DTIW) Locator should be used as much as possible (see DTIC-specific issues).

Technical.

Links to proprietary formats or large documents should have appropriate annotations (see Style Section).

II. Navigation/Organization

Dead Links.

No dead links are ever permitted to documents on the DTIC server. Links to external servers will be maintained as best as possible; the DTIC WWW server administrators will employ automated tools for checking for dead external links.

It is possible that a project manager or staff person may make changes to their documents which potentially could lead to dead links from other documents on the DTIC server. Proper coordination is required to avoid this (see DTIC-specific Issues).

* Upward Navigation.
    Documents should be designed so that users will have to rely (as little as possible) on navigational aids in the clients (e.g., back, forward buttons, history lists). For example, on a given (HTML) document, if two clicks of the "Back" button or command does not return a user to the DTIW Home Page or other Project Home Page, then there should be an explicit link to the DTIW Home Page or Project Home Page on that document. This is to provide users with an easy way to re-start or re-orient themselves.

* Restricted Access Projects.
    Documents and document sets that are available only to a restricted set of users will not be referenced from, or linked to/from, public access documents, or from document sets with different (i.e., more limited) access restrictions, without an explicit warning and information on the access restrictions.

Server-wide Organization.
    The DTIW will serve as the starting point for access to individual information sources on the DTIC server. The DTIC Home Page is organized around the structure of DTIC as an organization, and for information about DTIC as an organization. See the Style Section for information on using icons for linking to these starting points.

Department of Defense-Wide Organization
    The DTIC http server is the host to the Department of Defense Home Page, DefenseLINK. It is therefore important that other services on the DTIC server link to DefenseLINK consistently. See the Style Section for information on using icons for linking.

    The DoD Home Page should be referred to as DefenseLINK (one word, with all capital letters spelling LINK).

    Organizational Home Pages should provide links to the organizations immediately above and below that organization; Organizational references to the Department of Defense should link to DefenseLINK.

III. Style

A. Interface

Important Icons: DTIW Home Page, DTIC Home Page, DefenseLINK Home Page.
    There is a standard icon for DTIW Home Page. This text string/URL can be used to display it (with link):

    <A   HREF="/dtiw"><IMG   alt="DEFENSE   TECHNICAL   INFORMATION   WEB" SRC="/icons/dtiw_icon.gif"></A>

There is a standard icon for DTIC Home Page. This text string/URL can be used to display it (with link):

<A HREF="/index.html"><IMG alt="DTIC Home Page" SRC="/icons/dtichp_icon.gif"></A>

The for a standard icon for DefenseLINK, use:

<A HREF="/defenselink"><IMG alt="DefenseLINK" SRC="/defenselink/icons/deflink_icon .gif"></A>

Related icons for other services may also be appropriate to use, see the /icons or /defenselink/icons, /airforcelink/icons, etc. directories for additional icons.

Project Home Pages on the DTIW server will include the DTIW Home Page icon and link. It is recommended, but not required, that the DTIW Home Page link also be included at other higher-level documents in a document set.

Project Home Page Identification.
Project Home Pages will use either a consistent icon or consistent naming for linking to that home page.

Project Status.
The phrase "under construction" will not be used to describe a document or document collection; DTIC does not make available systems and services which are not ready to be used. However, notes about the current status of a certain feature, or information about anticipated changes in a feature or document collection, are appropriate.

Hypertext Don'ts.
Link text which interferes with readability should be avoided. Specifically, meaningless words, like the phrases "click here," "select a link below" and "return to" will be avoided.

*Phrases like "back to," and "up to" and "on to" should be used only for specific, highly related sets of documents.

*Instructions to "click" an icon will not be used. Instructions which make assumptions about what or where a user has just viewed should be avoided.

* Icons.
Images primarily used as "buttons" or icons should be able to stand on their own, i.e., they should be recognizable and understandable without accompanying text. They should also be as small as possible, while still communicating their purposes. Icons used as buttons on the DTIC server will normally by 76x76 pixels.

Images.
Images will not be used indiscriminately and will be appropriate to the content. Using images/icons to create consistent and recognizable style to a set of documents is appropriate and encouraged.

* Metaphors.
The use of metaphors to organize information or sets of documents is allowed, e.g., a book metaphor, with a Table of Contents, or a building metaphor with various "rooms".

Titles.

All documents will have titles. Document titles will be as short as possible, but fully informative and specific, e.g., "DTIC Conference Schedule" is preferable to "Conference Schedule."

* Headers.

The use of HTML headers, e.g., <H1> will be appropriate to the information, i.e., they will generally be less than 1-2 lines of text. Paragraphs of any header levels are not allowed.

* Other HTML Markups.

HTML documents on the DTIC WWW server will take full advantage of all available features in the HTML standard in order to make each document as readable and usable as possible. Project Implementors are reminded that the HTML "Standard" is not fully implemented in all browsers; projects should use markup appropriate to the capabilities of the browsers used by the target user community.

Real Information vs. Technical Advice.

Information of a technical nature (vs. real information) about a given document or set of documents (e.g., advice on which clients to use), will be segregated in separate documents. If the information is brief (See Technical Style Section) it will be marked using <EM> (emphasis) or <STRONG> (strong emphasis).

B. Technical

Large Files.

Links to individual files or collections of links to files larger than 100,000K will include an explicit note of the file size. The note will be marked using <EM> (emphasis) or <STRONG> (strong emphasis).

Proprietary File Types.

Links and CGI/gateways to individual files or collections of links to files in proprietary formats (e.g., MS Word, Power Point, etc.) will be explicitly noted. The note will be marked using <EM> (emphasis) or <STRONG> (strong emphasis).

The addition of non-standard MIME types which are recognized by the DTIC server will be coordinated with DTIC -Z.

URL Styles.

Relative URLs (for both HREF= and IMG SRC= values) are to be used whenever possible, in order to keep projects portable. For example, /defenselink/searchpage.html is preferable to http://www.dtic.dla.mil/defenselink/searchpage.html

* Images.

Imbedded images are to be kept to as small a size as possible. They must not be used indiscriminately and will be appropriate to the content. Image markup will use the alt=<text> syntax for usability with character (e.g., lynx) WWW clients. Images which are likely to be used in more than one project on the DTIC server (e.g., the DTIC logo) should be placed in the /icons directory for sharing.

* Image Maps.

It is strongly recommended that ISMAP (image maps) contain corresponding text links available to users. ALT statements for imagemaps will include the text "Imagemap" to inform users who

have not (or cannot) load images that the image is linked to the imagemap program (which they may not be able to use) and not another document.

Standard Mail Form Program.
Comment/order E-mail forms which simply return sets of field name=value (i.e., most mail forms) will use this link in the action statement:

/cgi-bin/mail-fields.pl

URL Names.
For Project Home Pages, URLs will use simple, understandable words, and be kept as short as possible, (i.e., the index.html standard, which allows a directory name to be used, will be employed). For example, http://www.dtic.dla.mil/project/ is preferable to http://www.dtic.dla.mil/project/project_home.html.

URL Case.
All URLs and file names will use lower case, unless Upper Case is required and justified. All file name extensions will also be in lower case. For example, http://www.dtic.dla.mil/icons/mypicture.GIF and http://www.dtic.dla.mil/MyHomePage.html are both prohibited.

Client Instructions.
Instructions specific to an individual WWW client should be avoided. If they are included, it should be described as an example, and not as a generic instruction.

* Feature Variations.
Document authors should be aware of the variations in features available on different clients and author their documents accordingly. Project staffers are encouraged to be familiar with what is available and what is being used in order to avoid being too conservative, or too liberal, in terms of implementing new functionality. For example, implementing the "mailto:" URL is not necessarily a bad thing (even if it is not widely used) if an E-mail address is explicitly included as well. The use of character clients is also an important consideration.

* Content vs. Format.
Document authors should recognize that HTML is as much a method for organizing information by content and structure as it is a formatting language. HTML should be used to structure the content of documents, as well as to format them. For example, headers should be in numeric order if possible; similar documents and elements within those documents should be marked up in a consistent manner.

IV. DTIC-specific issues

* DTIC Locator.
The DTIW Locator shall be the primary method for creating lists of external links. More specific lists, or lists outside the scope of the DTIW locator are permitted, but they must be kept up to date, a statement explaining their purpose will be included, and they must not duplicate a category in the Locator. The DTIW Locator is expandable and can accommodate new categories or information.

Coordination.

Project implementors should coordinate in advance with DTIC-Z staff and other WWW project implementors, when a new project is made available, or substantial changes are made to existing projects.

Public Release.
Project implementors are responsible for documenting that information made available in unrestricted directories is authorized for public release.

Compliance.
Project leaders and staff working on those projects are directly responsible for compliance with these standards, or for recommending changes in items which are in conflict with the goals of their projects.

Quality Control.
Project leaders are responsible for ensuring the quality and functionality of their HTML documents, forms and related CGI programs. Each Project Home Page, in addition to having a project leader, will have one or more "Quality Testers."

The Quality Tester(s) will review all changes to projects on the DTIC development server prior to their loading on the operational server. Project leaders and staff are responsible for ensuring review by the Quality Tester(s) prior to files being moved to the operational server.

The Quality Tester(s) will also be responsible for reviewing a project against the existing DTIC WWW Standards and Guidelines. If the requirements of a project are in conflict with these standards, Project leaders will submit an amendment to the standards, to DTIC-D. This amendment will document the new or revised standard.

* Single Source Policy
Project leaders should develop procedures (both manual and automatic) which minimize the chances of errors in source documents and which increase the efficiency by which documents are made available on the web. This is especially important when documents are created in other formats and must be converted into HTML (or other format) for use on the net.

Procedures.
A separate procedures document will provide details on the specific steps to be taken to use the DTIC WWW server for publishing electronic information.

V. Additional recommendations and points to ponder

* Developmental Testing.
In addition to review by Quality Testers for operational projects, it is strongly suggested that disinterested testers be employed to thoroughly check the features and contents in a project or document as the project is developed and as part of the development process. Testers can be found in other DTIC offices/directorates.

* Reinventing the Wheel.
Project implementors are encouraged to explore the Internet to review servers which might have done similar things as their projects. They are also encouraged to create processes and procedures which may be used in other situations at DTIC in the future.

* Training.

DTIC staff are encouraged to obtain training and information necessary to implement their WWW projects.

* Usage Reports.

Project leaders should review and analyze the usage reports on their documents and document collections, and use this information to improve their services.

* The Changing WWW environment

Project leaders are reminded that the WWW/HTML environment is changing rapidly; the number of users and servers is growing tremendously; new software for browsing, authoring, converting, serving and searching becomes available every week. The sources for these new products are both commercial and from those making their products freely available. Project leaders should from time-to-time evaluate the needs of their projects against the changing environment and make adjustments accordingly.

* Continuous Improvement

Project leaders should take advantage the of the "live" nature of WWW services by continuously improving their services. While frivolous and arbitrary change may be distracting to users, improvements to services based on user feedback, new ideas for organization, new HTML features, etc. will keep services fresh and will provides users reasons to take another look.

D.2 Example of GILS-Compliant WWW Page from DTIC

DEFENSE TECHNICAL INFORMATION CENTER

Acronym: DTIC

Originator: Defense Technical Information Center

Control_Identifier: DTIC-01

Abstract: The Defense Technical Information Center has a DoD-wide responsibility for collecting, analyzing, and disseminating reports and descriptions of research projects performed by or under contract to all parts of DoD and for serving the Scientific and Technical Information (STI) needs of DoD, according to the directives governing the DoD Scientific and Technical Information Program.

Format: Bibliographies/Full Text

Purpose: The Defense Technical Information Center is designated to provide a source of Scientific Technical Information Program (STIP) services to assist in carrying out STIP policy and administration; to perform technical information support services for the Office of the UnderSecretary of Defense (Acquisition & Technology) and OSD Principal Staff Assistants; to operate DoD-wide STI systems; to act as a central coordinating point for DoD STI data bases and systems; and to explore and demonstrate new supporting technology.

Access_Constraints: DTIC, with its holdings of classified, limited, and unclassified/unlimited data, serves only the Defense research community. Among those eligible to receive DTIC services are: Components of the Department of Defense; Government, Libraries and information centers; DoD military and civilian students and Universities involved in federally funded research throughout the United States; and Government contractors. All users are required to register for DTIC services. A registration packet with necessary forms and information about DTIC products and services is available upon request. Both government and contractor organizations must submit a completed DD Form 1540 Registration for Scientific and Technical Information Services. Contractor organizations must complete a separate DD Form 1540 for each contract or grant. Basic registration allows contractors to receive unclassified/unlimited or classified or otherwise limited data and services as required.

Use_Constraints: DTIC's unclassified/unlimited technical reports and bibliographic information is available to the General Public through the National Technical Information Service (NTIS). DTIC documents released to NTIS are indexed in NTIS's Government Reports Announcements and Index and citations to them are available on-line through the NTIS Bibliographic Data File.

Availability: Distributor:
Name: UnderSecretary of Defense (Acquisition & Technology)
Organization: Defense Technical Information Center
Street_Address: Cameron Station, Building 5
City: Alexandria
State: VA
Zip_Code: 22304-6145
Country: USA
Telephone: (703) 274-6871
Order_Process: The general public can order DTIC's unclassified/unlimited

Data & Analysis Center for Software

technical reports from the National Technical Information Service (NTIS)
Telephone: (703) 487-4650
Fax_Number: (703) 321-8547

Point_of_Contact: Name: DTIC Registration
Organization: Defense Technical Information Center
Street_Address: Cameron Station, Building 5
City: Alexandria
State: VA
Zip_Code: 22304-6145
Telephone: (703) 274-6871
Network_Address: reghelp@dgis.dtic.dla.mil
Fax: (703) 274-9307
Hours_of_Service: 6:00 a.m. - 5:30 p.m.

Record_Source: Defense Technical Information Center (DTIC)

Date_Last_Modified: June 1994
_____

Document was updated on:
Today's date and time:
STINET WWW
stinet@dgis.dtic.dla.mil

APPENDIX E HTML LANGUAGE FEATURES

The purpose of this appendix is to provide a reference for the hypertext markup language. It is not intended to be a comprehensive tutorial or "HTML programmer's guide," rather to provide sufficient detail for a Web author or maintainer to be able to interpret and/or update the HTML markup in a file.

The first section of the appendix is a Quick Reference, which lists the most common tags, and provides a few examples. The second section, referred to as a "Test Pattern," was constructed to provide a complete set of HTML tags, at HTML Level 2. The section itself is marked up in HTML, and so it can be used to investigate how any browser and platform combination will interpret the tags and display the file. The third section is an example of how the Test Pattern file looks in one environment, specifically Mosaic Version 2.0 for X Windows (Unix) on a monochrome monitor. The displayed page was saved to a Postscript file, which was then printed for inclusion in the handbook.

E.1 Quick Reference

Quick Reference Guide to Common HTML Tags
Prepared by Brian Davies, Kaman Sciences Corporation

This document is intended to be a quick reference guide to HTML tags. It was originally compiled to serve as a reference guide to HTML tags without any explanation of the function of the tags. Some examples have been included, where needed, to show the use syntax of HTML tags.

This document is *not* intended to serve as an introduction to HTML scripting. There are several excellent documents already in existence for that purpose. This document does not include any tags which are considered Netscape™ Enhanced tags because these tags are not currently supported by all browsers.

For more information about developing WWW resources, check out the offerings at the University of Toronto: (http://www.utirc.utoronto.ca/HTMLdocs/intro_tools.html), and the University of Washington: (http://www.uwtc.washington.edu/Computing/WWW/UWWeb.html).

HTML tags are typically used in the following format:

<TAG>your text or images here</TAG>

Note: HTML tags are case-insensitive. <TITLE> is the same as <title> or <TiTle>

Open statement	Close statement	Tag function
General Use Tags		
<!--	-->	Note space which will not appear when the html page is displayed
<HTML>	</HTML>	Needed for compatibility with some older browsers
<HEAD>	</HEAD>	Used to defined the Head section of an HTML document. Not needed for most current documents
<BODY>	</BODY>	Used to define the Body section of an HTML document. Not needed for most current documents
<TITLE>	</TITLE>	Title a document
<H1> thru <H6>	</H1> thru </H6>	Headings (H1 is largest, H6 smallest)
<P>		Paragraph break
 		Line break
<HR size=n>		Horizontal Rule - n is the size of the rule (not needed for a simple bar)
<P ALIGN=CENTER>		Centered Paragraph

Data & Analysis Center for Software

Links

<A NAME="???">	</A>	Anchor inside document (internal link)
<A HREF="???">	</A>	Anchor to other files (external link)

??? can be a number of different types of links including ftp, http, a local file, mailto, or another spot in the same document.

Examples
Example of a NAME tag in a document:
<A NAME="software">Software Engineering Tools</A>

A link to a specific spot in a document:
<A HREF="tool.html#software">Software Engineering Tools</A>

Link to another http site:<A HREF="http://www.utica.kaman.com">Check out the DACS here</A>

Link to another document on the same server (using relative paths):
<A HREF="../home_page.html">My home page</A>

Using a mailto link:
<A HREF="mailto:auser@utica.kaman.com">Send me mail</A>

Using a link to ftp a file:
<A HREF="ftp:">Get the file here</A>

Images

<IMG ALIGN=x SRC="filename" ALT="text">	Add image to page with ALTernate name provided for users with text-only browsers x allows you to align text either TOP, MIDDLE, or BOTTOM

Examples
Example of an image tag:
<IMG ALIGN=MIDDLE SRC="DACS_home.gif" ALT="DACS Home Image">

Example of an image used as a link:
<A HREF="index.html"><IMG ALIGN=MIDDLE SRC="DACS_home.gif" ALT="DACS Home Image"></A>

Lists

<UL>	</UL>	Unordered list (items will appear bulleted)
<LI>		List item
<OL>	</OL>	Ordered list (items will

		appear in a numbered list)
<LI>		List item
<DL>	</DL>	Definition list
<DT>		Definition term

Examples
Example of an Unordered list entry:
System Requirements
```
<UL>
<LI> 64 Mb of RAM
<LI> 200 Mb of Disk Space
<LI> Color Monitor
</UL>
```

Example of Ordered list entry

```
<OL>
<LI> Add two cups flour and three eggs
<LI> Mix in 1 tbsp. Vanilla and 1 cup brown sugar
<LI> Bake for 30 minutes at 350 degrees
</OL>
```

Example of an Definition list
```
<DL>
<DT> Name:
<DT> Address:
</DL>
```

Formatting

<PRE>	</PRE>	Use pre-formatted text (useful for dealing with tables and columns or including "computer" text)
<BLOCKQUOTE>	</BLOCKQUOTE>	Blockquote text
<B>	</B>	Boldface
<I>	</I>	Italics
<TT>	</TT>	Typewriter font
<EM>	</EM>	Emphasis (italics, or plain if surrounding text is already italicized)
<STRONG>	</STRONG>	Strong emphasis (bold)
<ADDRESS>	</ADDRESS>	Specify address information such as author

E.2 Test Pattern

This is HTML Level 2.
<HTML>
<HEAD>
<TITLE>Test Pattern</TITLE>
</HEAD>
<BODY>
<P>
<H1><A NAME="top">HTML EXAMPLES</H1>
<P>
This document contains a wide range of HTML elements. It can be used to investigate your browser's characteristics. Although case is not specified for HTML elements, all the examples here use uppercase to set the HTML code off from the surrounding text.
<P>
<UL>
<LI><A HREF="#headings">Headings</A>
<LI><A HREF="#logical-styles">Logical Styles</A>
<LI><A HREF="#physical-styles">Physical Styles</A>
<LI><A HREF="#text-formatting">Some Special Text Formatting Modes</A>
<LI><A HREF="#line-breaks">Paragraphs and Line Breaks</A>
<LI><A HREF="#special-characters">Special Characters</A>
<LI><A HREF="#comments">Comments</A>
<LI><A HREF="#lists">Lists</A>
<LI><A HREF="#anchors">Anchors</A>
<LI><A HREF="#images">Included Images</A>
</UL>
<P>

<H1><A NAME="headings">Headings</A></H1>
<P>
Headings are created by enclosing the heading text in the HTML elements <<CODE>Hnn</CODE>> and <<CODE>/Hnn</CODE>> where <CODE>nn</CODE> represents the heading level. Six levels of headings are defined:
<H1>Main Heading</H1>
<H2>2nd Level Heading</H2>
<H3>3rd Level Heading</H3>
<H4>4th Level Heading</H4>
<H5>5th Level Heading</H5>
<H6>6th Level Heading</H6>
<H7>7th Level Heading (Not defined)</H7>
<P>
Did you know a heading can also be an anchor, e.g., a link?
<H1><A HREF="#line-breaks">A Link to Line Breaks Below</A></H1>

<H1><A NAME="logical-styles">Logical Styles</A></H1>
<UL>
<LI>Text is <EM>emphasized</EM> by the <<CODE>EM</CODE>> element.
<LI>Strong <STRONG>emphasis</STRONG> is provided by the <<CODE>STRONG</CODE>> element.

<LI>Source code, e.g., <CODE>printf("hello world\n")</CODE>, should be constructed from the <<CODE>CODE</CODE>> element.
<LI>User input, e.g., <KBD>your username</KBD>, should be constructed from the <<CODE>KBD</CODE>> element.
<LI>A variable name, e.g., <VAR>foobar</VAR>, should be constructed from the <<CODE>VAR</CODE>> element.
<LI>Definitions, e.g., <DFN>A definition</DFN>, should be constructed from the <<CODE>DFN</CODE>> element.
<LI>The citation <CITE>The Elements of Style</CITE> is constructed from the <<CODE>CITE</CODE>> element.
</UL>
<P>
The above elements must be balanced by closing elements, e.g.,
<<CODE>/EM</CODE>>,
<<CODE>/STRONG</CODE>>,
<<CODE>/CODE</CODE>>,
<<CODE>/KBD</CODE>>,
<<CODE>/VAR</CODE>>,
<<CODE>/DFN</CODE>>, and
<<CODE>/CITE</CODE>>.

<H1><A NAME="physical-styles">Physical Styles</A></H1>
<P>
The use of physical styles is not in the <EM>spirit</EM> of HTML.
<UL>
<LI><B>Fixed width text</B> is created by the <<CODE>T</CODE>> element.
<LI><B>Boldface</B> is created by the <<CODE>B</CODE>> element.
<LI><I>Italics</I> is created by the <<CODE>I</CODE>> element.
<LI><U>Underlined Text</U> is created by the <<CODE>U</CODE>> element.
</UL>
<P>
The above elements must also be balanced by closing elements, e.g.,
<<CODE>/TT</CODE>>,
<<CODE>/B</CODE>>,
<<CODE>/I</CODE>>, and
<<CODE>/U</CODE>>.

<H1><A NAME="text-formatting">Some Special Text Formatting Modes</A></H1>
<P>
A long quote can be set off from the main text by enclosing the text of the quote between the <<CODE>BLOCKQUOTE</CODE>> and <<CODE>/BLOCKQUOTE</CODE>> elements. A comment by E. B. White illustrates:
<BLOCKQUOTE><B>Work from a suitable design.</B> Before beginning to compose something, gauge the nature and extent of the enterprise and work from a suitable design. Design informs even the simplest structure, whether of brick and steel or of prose. You raise a pup tent from one sort of vision, a cathedral from another...</BLOCKQUOTE>
<P>
Multiple blanks and carriage returns are significant in <EM>preformatted</EM> text. Preformatted text is enclosed in <<CODE>PRE</CODE>> and <<CODE>/PRE</CODE>>:
<PRE WIDTH=60>
    Said studious Robbie,
    "To all it's plain to see,

```
      Oddball spelling
      Tense no telling,
    English is Greek to me."
</PRE>
```
The width of preformatted text can be specified by the WIDTH attribute in the <CODE>PRE</CODE> element, e.g., <<CODE>PRE WIDTH=60</CODE>>

```
<P>
```
An address is enclosed in <<CODE>/ADDRESS</CODE>> and <<CODE>/ADDRESS</CODE>>:

```
<ADDRESS>
Mr. John Doe
123 Main Street
Anytown, USA
(999) 999-9999
jdoe@internet.address.com
</ADDRESS>
```
The above address appears on one line with Mosaic version 2.0 for X Windows. The following is preformatted text enclosed within address elements:

```
<ADDRESS><PRE>
Mr. John Doe
123 Main Street
Anytown, USA
(999) 999-9999
jdoe@internet.address.com
</PRE></ADDRESS>
```
The next example is an address enclosed with preformatted text elements:

```
<PRE><ADDRESS>
Mr. John Doe
123 Main Street
Anytown, USA
(999) 999-9999
jdoe@internet.address.com
</ADDRESS></PRE>
```

```
<H1><A NAME="line-breaks">Paragraphs and Line Breaks</A></H1>
<P>
```
Carriage returns are without significance in HTML (except in preformatted text). The line break element <<CODE>BR</CODE>> indicates a new line. Here is the above address rendered with line breaks, instead of preformatted text:

```
<ADDRESS>
Mr. John Doe<BR>123 Main Street<BR>Anytown, USA<BR>
(999) 999-9999<BR>
jdoe@internet.address.com<BR>
</ADDRESS>
```
The <<CODE>BR</CODE>> element is empty and requires no closing element.

```
<P>
```
Paragraphs are indicated by the <<CODE>P</CODE>> element. HTML+ will accept an optional close paragraph element <<CODE>/P</CODE>>. HTML+ will also allow a paragraph to be marked for use in an URL and to have its alignment specified (center, left, right, justify, indent).

```
<P>
```
A horizontal line is created by the <<CODE>HR</CODE>> element:

```
<HR>
```

The <<CODE>HR</CODE>> element is empty and requires no closing element.

<H1><A NAME="special-characters">Special Characters</A></H1>
<UL>
<LI> <, less than, <CODE>&lt;</CODE>
<LI> >, greater than, <CODE>&gt;</CODE>
<LI> &, ampersand, <CODE>&amp;</CODE>
<LI> ", double quote, <CODE>&quot;</CODE>
<LI> , non-breaking space, <CODE>&nbsp;</CODE>
<P>
<LI> Æ, uppercase AE diphthong, <CODE>&AElig;</CODE>
<LI> Á, uppercase A with acute accent, <CODE>&Aacute;</CODE>
<LI> Â, uppercase A with circumflex accent, <CODE>&Acirc;</CODE>
<LI> À, uppercase A with grave accent, <CODE>&Agrave;</CODE>
<LI> Å, uppercase A with ring, <CODE>&Aring;</CODE>
<LI> Ã, uppercase A with tilde, <CODE>&Atilde;</CODE>
<LI> Ä, uppercase A with umlaut, <CODE>&Auml;</CODE>
<LI> Ç, uppercase C with cedilla, <CODE>&Ccedil;</CODE>
<LI> Ð, Icelandic uppercase Eth, <CODE>&ETH;</CODE>
<LI> É, uppercase E with acute accent, <CODE>&Eacute;</CODE>
<LI> Ê, uppercase E with circumflex accent, <CODE>&Ecirc;</CODE>
<LI> È, uppercase E with grave accent, <CODE>&Egrave;</CODE>
<LI> Ë, uppercase E with umlaut, <CODE>&Euml;</CODE>
<LI> Í, uppercase I with acute accent, <CODE>&Iacute;</CODE>
<LI> Î, uppercase I with circumflex accent, <CODE>&Icirc;</CODE>
<LI> Ì, uppercase I with grave accent, <CODE>&Igrave;</CODE>
<LI> Ï, uppercase I with umlaut, <CODE>&Iuml;</CODE>
<LI> Ñ, uppercase N with tilde, <CODE>&Ntilde;</CODE>
<LI> Ó, uppercase O with acute accent, <CODE>&Oacute;</CODE>
<LI> Ô, uppercase O with circumflex accent, <CODE>&Ocirc;</CODE>
<LI> Ò, uppercase O with grave accent, <CODE>&Ograve;</CODE>
<LI> Ø, uppercase O with slash, <CODE>&Oslash;</CODE>
<LI> Õ, uppercase O with tilde, <CODE>&Otilde;</CODE>
<LI> Ö, uppercase O with umlaut, <CODE>&Ouml;</CODE>
<LI> Þ, Icelandic uppercase Thorn, <CODE>&THORN;</CODE>
<LI> Ú, uppercase U with acute accent, <CODE>&Uacute;</CODE>
<LI> Û, uppercase U with circumflex accent, <CODE>&Ucirc;</CODE>
<LI> Ù, uppercase U with grave accent, <CODE>&Ugrave;</CODE>
<LI> Ü, uppercase U with umlaut, <CODE>&Uuml;</CODE>
<LI> Ý, uppercase Y with acute accent, <CODE>&Yacute;</CODE>
<P>
<LI> æ, lowercase ae diphthong, <CODE>&aelig;</CODE>
<LI> á, lowercase a with acute accent, <CODE>&aacute;</CODE>
<LI> â, lowercase a with circumflex accent, <CODE>&acirc;</CODE>
<LI> à, lowercase a with grave accent, <CODE>&agrave;</CODE>
<LI> å, lowercase a with ring, <CODE>&aring;</CODE>
<LI> ã, lowercase a with tilde, <CODE>&atilde;</CODE>
<LI> ä, lowercase a with umlaut, <CODE>&auml;</CODE>
<LI> ç, lowercase c with cedilla, <CODE>&ccedil;</CODE>
<LI> ð, Icelandic lowercase eth, <CODE>&eth;</CODE>
<LI> é, lowercase e with acute accent, <CODE>&eacute;</CODE>
<LI> ê, lowercase e with circumflex accent, <CODE>&ecirc;</CODE>

```
<LI> &egrave;, lowercase e with grave accent, <CODE>&amp;egrave;</CODE>
<LI> &euml;, lowercase e with umlaut, <CODE>&amp;euml;</CODE>
<LI> &iacute;, lowercase i with acute accent, <CODE>&amp;iacute;</CODE>
<LI> &icirc;, lowercase i with circumflex accent, <CODE>&amp;icirc;</CODE>
<LI> &igrave;, lowercase i with grave accent, <CODE>&amp;igrave;</CODE>
<LI> &iuml;, lowercase i with umlaut, <CODE>&amp;iuml;</CODE>
<LI> &ntilde;, lowercase n with tilde, <CODE>&amp;ntilde;</CODE>
<LI> &oacute;, lowercase o with acute accent, <CODE>&amp;oacute;</CODE>
<LI> &ocirc;, lowercase o with circumflex accent, <CODE>&amp;ocirc;</CODE>
<LI> &ograve;, lowercase o with grave accent, <CODE>&amp;ograve;</CODE>
<LI> &oslash;, lowercase o with slash, <CODE>&amp;oslash;</CODE>
<LI> &otilde;, lowercase o with tilde, <CODE>&amp;otilde;</CODE>
<LI> &ouml;, lowercase o with umlaut, <CODE>&amp;ouml;</CODE>
<LI> &szlig;, German lowercase sharp s, <CODE>&amp;szlig;</CODE>
<LI> &thorn;, Icelandic lowercase thorn, <CODE>&amp;thorn;</CODE>
<LI> &uacute;, lowercase u with acute accent, <CODE>&amp;uacute;</CODE>
<LI> &ucirc;, lowercase u with circumflex accent, <CODE>&amp;ucirc;</CODE>
<LI> &ugrave;, lowercase u with grave accent, <CODE>&amp;ugrave;</CODE>
<LI> &uuml;, lowercase u with umlaut, <CODE>&amp;uuml;</CODE>
<LI> &yacute;, lowercase y with acute accent, <CODE>&amp;yacute;</CODE>
<LI> &yuml;, lowercase y with umlat, <CODE>&amp;yuml;</CODE>
</UL>

<H1><A NAME="comments">Comments</A></H1>
<P>
Comment lines are enclosed in &lt;<CODE>!--</CODE> and <CODE>--</CODE>&gt;.

<!--This line is a comment.-->
<!--So is this line. They should not appear in your browser window.-->

<H1><A NAME="lists">Lists</A></H1>
<P>
The elements &lt;<CODE>UL</CODE>&gt; and &lt;<CODE>/UL</CODE>&gt; are used to group
nested paragraphs. The empty element &lt;<CODE>LI</CODE>&gt; prefaces a paragraph by a bullet:
<UL>
An unbulleted list element<BR>
Another unbulleted list element
<LI> A bulleted list element
<LI> Another bulleted list element
<UL>
<LI> A second level of bulleted items
<LI> Another bulleted item
<UL>
<LI> A third level<BR>
An unbulleted item at the third level
</UL>
</UL>
<LI> A bulleted item at the first level
</UL>
<P>
&lt;<CODE>OL</CODE>&gt; and &lt;<CODE>/OL</CODE>&gt; group nested numbered
paragraphs. The empty element &lt;<CODE>LI</CODE>&gt; prefaces a paragraph by a numeral:
```

```
<OL>
An unnumbered list element<BR>
Another unnumbered list element
<LI> A numbered list element
<LI> Another numbered list element
<OL>
<LI> A second level of numbered items
<LI> Another numbered item
<OL>
<LI> A third level<BR>
An unnumbered item at the third level
</OL>
</OL>
<LI> A numbered item at the first level
</OL>
<P>
```

<<CODE>MENU</CODE>> lists are similar to <<CODE>UL</CODE>> lists, but are formatted more compactly. <<CODE>DIR</CODE>> lists have elements arranged in columns across the page. The elements <<CODE>DL</CODE>>, <<CODE>DT</CODE>>, <<CODE>DD</CODE>>, and <<CODE>/DL</CODE>> are used to construct glossary lists. List types can be nested at different levels:

```
<OL>
An unnumbered list element<BR>
Another unnumbered list element
<LI> A numbered list element
<LI> Another numbered list element
<UL>
<LI> A second level of list items
<LI> Another bulleted item
<UL>
<LI> A third level<BR>
An unbulleted item at the third level
</UL>
</UL>
<LI> A numbered item at the first level
</OL>


<H1><A NAME="anchors">Anchors</A></H1>
<P>
```

Anchors define hypertext links and their attributes. The format of an anchor in HTML is
```
<P>
<CODE>&lt;A one or more attributes&gt;anchor text&lt;/A&gt;</CODE>
<P>
```
The <CODE>HREF</CODE> attribute is used to create a link. The format of an anchor containing an <CODE>HREF</CODE> attribute is as follows:
```
<P>
<CODE>&lt;A HREF=&quot;URL&quot;&gt;anchor text&lt;/A&gt;</CODE>
<P>
```
where a URL is a Universal Resource Locator. For example, the HTML for a link to the Yahoo virtual library would look like this:
```
<P>
<CODE>&lt;A HREF=&quot;http://www.yahoo.com/&quot;&gt;<BR>
```

Yahoo - A Guide to WWW </A></CODE>
<P>
The above code should be rendered by your browser as a clickable link:
<A HREF="http://www.yahoo.com/">Yahoo - A Guide to WWW </A>.
URLs can be relative to a given document and can point to a named location in a given document. Given that this document is located at <VAR>http://www.utica.kaman.com/internet/handbook/test-pattern.html</VAR>, the following two links are equivalent:
<P>
<A   HREF="#headings"><CODE><A   HREF="#headings">Headings</A> </CODE></A>
<P>
<A HREF="http://www.utica.kaman.com/internet/handbook/test-pattern.html#headings"><CODE> <A HREF="http://www.utica.kaman.com/internet/handbook/
 test-pattern.html#headings">Headings</A></CODE></A>
<P>
A location in a document is currently defined by the <CODE>NAME</CODE> attribute in an anchor. (<CODE>NAME</CODE> will be replaced by the paragraph attribute <CODE>ID</CODE> in HTML+.) For example, the HTML code defining the location in this document pointed to by the above links is as follows:
<P>
<CODE><H1><A
NAME="headings">Headings</A></H1></CODE>
<P>
(Since this location is a header, the anchor is enclosed in <CODE><H1></CODE> and <CODE></H1></CODE>.)
<P>
There are some other lesser known attributes for anchors:
<CODE>REL</CODE>, <CODE>REV</CODE>, <CODE>URN</CODE>, <CODE>TITLE</CODE>, and <CODE>METHODS</CODE>.

<H1><A NAME="images">Included Images</A></H1>
<P>
Text can be wrapped around images in different locations:

<IMG SRC="http://www.utica.kaman.com/awareness/newsletters/capital.gif"
ALIGN=BOTTOM
ALT="The Capitol Dome"></A>Caption at bottom

<IMG SRC="http://www.utica.kaman.com/awareness/newsletters/capital.gif"
ALIGN=MIDDLE
ALT="The Capitol Dome"></A>Caption in Middle

<IMG SRC="http://www.utica.kaman.com/awareness/newsletters/capital.gif"
ALIGN=TOP
ALT="The Capitol Dome"></A>Caption on Top

<A HREF="#top">
<IMG SRC="http://www.utica.kaman.com/awareness/newsletters/capital.gif">
The image and the caption are links</A>
</BODY>
</HTML>

E.3 Formatted
HTML EXAMPLES

This document contains a wide range of HTML elements. It can be used to investigate your browsers's characteristics. Case is not specified for HTML elements, but all examples use uppercase to set the HTML code off from the surrounding text.

- Headings
- Logical Styles
- Physical Styles
- Some Special Text Formatting Modes
- Paragraphs and Line Breaks
- Special Characters
- Comments
- Lists
- Anchors
- Included Images

Headings

Headings are created by enclosing the heading text in the HTML elements `<Hnn>` and `</Hnn>`, where `nn` represents the heading level. Six levels of headings are defined:

Main Heading

2nd Level Heading

3rd Level Heading

4th Level Heading

5th Level Heading

6th Level Heading

7th Level Heading (Not defined)

Did you know a heading can also be an anchor, e.g., a link?

A Link to Line Breaks Below

Logical Styles

- Text is *emphasized* by the `<EM>` element.
- Strong **emphasis** is provided by the `<STRONG>` element.
- Source code, e.g., `printf("hello world\n")`, should be constructed from the `<CODE>` element.
- User input, e.g., `your username`, should be constructed from the `<KBD>` element.
- A variable name, e.g., *foobar*, should be constructed from the `<VAR>` element.
- Definitions, e.g., A definition, should be constructed from the `<DFN>` element.
- The citation *The Elements of Style* is constructed from the `<CITE>` element.

Data & Analysis Center for Software

The above elements must be balanced by closing elements, e.g., </EM>, </STRONG>, </CODE>, </KBD>, </VAR>, </DFN>, and </CITE>.

Physical Styles

The use of physical styles is not in the *spirit* of HTML.

- **Fixed width text** is created by the <T> element.
- **Boldface** is created by the <B> element.
- *Italics* is created by the <I> element.
- Underlined Text is created by the <U> element.

The above elements must also be balanced by closing elements, e.g., </TT>, </B>, </I>, and </U>.

Some Special Text Formatting Modes

A long quote can be set off from the main text by enclosing the text of the quote between the <BLOCKQUOTE> and </BLOCKQUOTE> elements. A comment by E. B. White illustrates:

> **Work from a suitable design.** Before beginning to compose something, gauge the nature and extent of the enterprise and work from a suitable design. Design informs even the simplest structure, whether of brick and steel or of prose. You raise a pup tent from one sort of vision, a cathedral from another...

Multiple blanks and carriage returns are significant in *preformatted* text. Preformatted text is enclosed in <PRE> and </PRE>:

```
Said studious Robbie,
"To all it's plain to see,
   Oddball spelling
   Tense no telling,
English is Greek to me."
```

The width of preformatted text can be specified by the WIDTH attribute in the PRE element, e.g., <PRE WIDTH=60>

An address is enclosed in </ADDRESS> and </ADDRESS>:
*Mr. John Doe 123 Main Street Anytown, USA (999) 999-9999
jdoe@internet.address.com*
The above address appears on one line with Mosaic version 2.0 for X Windows. The following is preformatted text enclosed within address elements:

```
Mr. John Doe
123 Main Street
Anytown, USA
(999) 999-9999
jdoe@internet.address.com
```

The next example is an address enclosed with preformatted text elements:

*Mr. John Doe
123 Main Street
Anytown, USA
(999) 999-9999
jdoe@internet.address.com*

Paragraphs and Line Breaks

Carriage returns are without significance in HTML (except in preformatted text). The line break element `<BR>` indicates a new line. Here is the above address rendered with line breaks, instead of preformatted text:
Mr. John Doe
123 Main Street
Anytown, USA
(999) 999–9999
jdoe@internet.address.com
The `<BR>` element is empty and requires no closing element.

Paragraphs are indicated by the `<P>` element. HTML+ will accept an optional close paragraph element `</P>`. HTML+ will also allow a paragraph to be marked for use in an URL and to have its alignment specified (center, left, right, justify, indent).

A horizontal line is created by the `<HR>` element:

The `<HR>` element is empty and requires no closing element.

Special Characters

- <, less than, `<`
- >, greater than, `>`
- &, ampersand, `&`
- ", double quote, `"`
- , non–breaking space, ` `

- Æ, uppercase AE diphthong, `Æ`
- Á, uppercase A with acute accent, `Á`
- Â, uppercase A with circumflex accent, `Â`
- À, uppercase A with grave accent, `À`
- Å, uppercase A with ring, `Å`
- Ã, uppercase A with tilde, `Ã`
- Ä, uppercase A with umlaut, `Ä`
- Ç, uppercase C with cedilla, `Ç`
- Ð, Icelandic uppercase Eth, `Ð`
- É, uppercase E with acute accent, `É`
- Ê, uppercase E with circumflex accent, `Ê`
- È, uppercase E with grave accent, `È`
- Ë, uppercase E with umlaut, `Ë`
- Í, uppercase I with acute accent, `Í`
- Î, uppercase I with circumflex accent, `Î`
- Ì, uppercase I with grave accent, `Ì`
- Ï, uppercase I with umlaut, `Ï`
- Ñ, uppercase N with tilde, `Ñ`
- Ó, uppercase O with acute accent, `Ó`
- Ô, uppercase O with circumflex accent, `Ô`
- Ò, uppercase O with grave accent, `Ò`
- Ø, uppercase O with slash, `Ø`
- Õ, uppercase O with tilde, `Õ`
- Ö, uppercase O with umlaut, `Ö`
- Þ, Icelandic uppercase Thorn, `Þ`
- Ú, uppercase U with acute accent, `Ú`
- Û, uppercase U with circumflex accent, `Û`
- Ù, uppercase U with grave accent, `Ù`
- Ü, uppercase U with umlaut, `Ü`
- Ý, uppercase Y with acute accent, `Ý`

Data & Analysis Center for Software

- æ, lowercase ae diphthong, `æ`
- á, lowercase a with acute accent, `á`
- â, lowercase a with circumflex accent, `â`
- à, lowercase a with grave accent, `à`
- å, lowercase a with ring, `å`
- ã, lowercase a with tilde, `ã`
- ä, lowercase a with umlaut, `ä`
- ç, lowercase c with cedilla, `ç`
- ð, Icelandic lowercase eth, `ð`
- é, lowercase e with acute accent, `é`
- ê, lowercase e with circumflex accent, `ê`
- è, lowercase e with grave accent, `è`
- ë, lowercase e with umlaut, `ë`
- í, lowercase i with acute accent, `í`
- î, lowercase i with circumflex accent, `î`
- ì, lowercase i with grave accent, `ì`
- ï, lowercase i with umlaut, `ï`
- ñ, lowercase n with tilde, `ñ`
- ó, lowercase o with acute accent, `ó`
- ô, lowercase o with circumflex accent, `ô`
- ò, lowercase o with grave accent, `ò`
- ø, lowercase o with slash, `ø`
- õ, lowercase o with tilde, `õ`
- ö, lowercase o with umlaut, `ö`
- ß, German lowercase sharp s, `ß`
- þ, Icelandic lowercase thorn, `þ`
- ú, lowercase u with acute accent, `ú`
- û, lowercase u with circumflex accent, `û`
- ù, lowercase u with grave accent, `ù`
- ü, lowercase u with umlaut, `ü`
- ý, lowercase y with acute accent, `ý`
- ÿ, lowercase y with umlat, `ÿ`

Comments

Comment lines are enclosed in `<!--` and `-->`.

Lists

The elements `<UL>` and `</UL>` are used to group nested paragraphs. The empty element `<LI>` prefaces a paragraph by a bullet:

An unbulleted list element
Another unbulleted list element
- A bulleted list element
- Another bulleted list element
  - A second level of bulleted items
  - Another bulleted item
    - A third level
      An unbulleted item at the third level
- A bulleted item at the first level

`<OL>` and `</OL>` group nested numbered paragraphs. The empty element `<LI>` prefaces a paragraph by a numeral:

An unnumbered list element
Another unnumbered list element
1. A numbered list element
2. Another numbered list element

Data & Analysis Center for Software

1. A second level of numbered items
2. Another numbered item
    1. A third level
        An unnumbered item at the third level
3. A numbered item at the first level

<MENU> lists are similar to <UL> lists, but are formatted more compactly. <DIR> lists have elements arranged in columns across the page. The elements <DL>, <DT>, <DD>, and </DL> are used to construct glossary lists. List types can be nested at different levels:

An unnumbered list element
Another unnumbered list element
1. A numbered list element
2. Another numbered list element
    o A second level of list items
    o Another bulleted item
        • A third level
            An unbulleted item at the third level
3. A numbered item at the first level

Anchors

Anchors define hypertext links and their attributes. The format of an anchor in HTML is

```
<A one or more attributes>anchor text</A>
```

The HREF attribute is used to create a link. The format of an anchor containing an HREF attribute is as follows:

```
<A HREF="URL">anchor text</A>
```

where a URL is a Universal Resource Locator. For example, the HTML for a link to the Yahoo virtual library would look like this:

```
<A HREF="http://www.yahoo.com/">
Yahoo - A Guide to WWW </A>
```

The above code should be rendered by your browser as a clickable link: Yahoo – A Guide to WWW . URLs can be relative to a given document and can point to a named location in a given document. Given that this document is located at *http://www.utica.kaman.com/internet/handbook/test–pattern.html*, the following two links are equivalent:

```
<A HREF="#headings">Headings</A>
```

```
<A HREF="http://www.utica.kaman.com/internet/handbook/
test-pattern.html#headings">Headings</A>
```

A location in a document is currently defined by the NAME attribute in an anchor. (NAME will be replaced by the paragraph attribute ID in HTML+.) For example, the HTML code defining the location in this document pointed to by the above links is as follows:
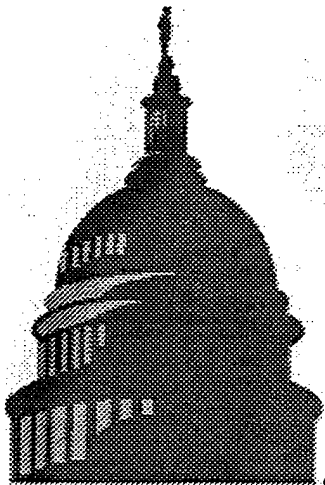
```
<H1><A NAME="headings">Headings</A></H1>
```

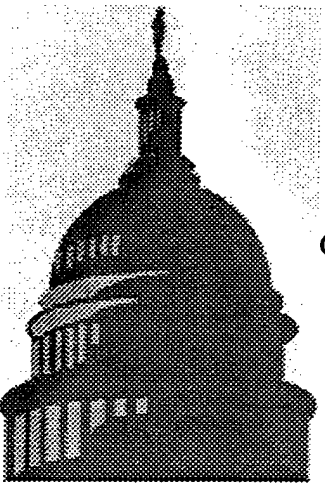(Since this location is a header, the anchor is enclosed in <H1> and </H1>.)

There are some other lesser known attributes for anchors: REL, REV, URN, TITLE, and METHODS.
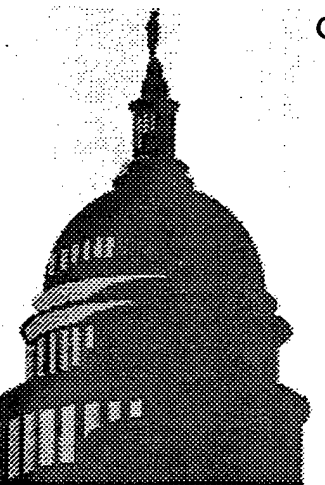
Included Images

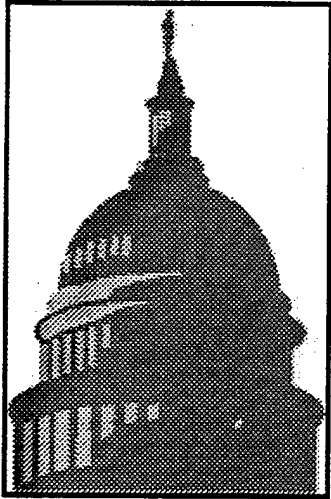Text can be wrapped around images in different locations:

Caption at bottom

Caption in Middle

Caption on Top

The image and the caption are links

Data & Analysis Center for Software